

Purple papers

Syntax and typed expression

MJos Predictive Model

The first manifestation of the Bangel Language

MJ Physics Engineering

Owned and operated by Michael Bangel

Clearwater, Florida

Recovered collection / terminology edition

300 recovered source pages

Prepared for review. Not published.

This source-faithful terminology edition updates the language name and publication identity while retaining mathematical expressions, source IDs, historical evidence and page graphics. It is not a fresh execution of the historical tests. Read the companion searchable library for current project sequencing and unresolved semantic relationships.

White papers are excluded. References to White paper identities remain historical citations. Original source hashes refer to the prior originals; revised output hashes are recorded separately.

Gold and Indigo: the original volumes were not recovered. Any new reconstruction is separately identified and must not be counted as those missing volumes.

Syntax expression for BP-ALG-001

```
module MJ.PBMEF.Blue.BP-ALG-001 {
  reference F-001;
  expression "F-001 = R0 -> R1";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-001: R0 -> R1

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-090 presents the normalized blueprint; Teal TP-001 expresses IF/HOW/THEN; Brown BR-001 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-001); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-001.

Syntax expression for BP-ALG-002

```
module MJ.PBMEF.Blue.BP-ALG-002 {
  reference F-002;
  expression "F-002 = R1 --validated--> R2";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-002: R1 --validated--> R2

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-052 presents the normalized blueprint; Teal TP-002 expresses IF/HOW/THEN; Brown BR-002 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-002); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-002.

Syntax expression for BP-ALG-003

```
module MJ.PBMEF.Blue.BP-ALG-003 {
  reference F-003;
  expression "F-003 = UNKNOWN != 0";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-003: UNKNOWN != 0

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-012 presents the normalized blueprint; Teal TP-003 expresses IF/HOW/THEN; Brown BR-003 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-003); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-003.

Syntax expression for BP-ALG-004

```
module MJ.PBMEF.Blue.BP-ALG-004 {
  reference F-004;
  expression "F-004 = m_floor = 23 mg";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-004: m_floor = 23 mg

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-076 presents the normalized blueprint; Teal TP-004 expresses IF/HOW/THEN; Brown BR-004 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-004); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-004.

Syntax expression for BP-ALG-005

```
module MJ.PBMEF.Blue.BP-ALG-005 {
  reference F-005;
  expression "F-005 = n_bar[t] = 0.9*n_bar[t-1] + 0.1*n[t]";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-005: $n_bar[t] = 0.9*n_bar[t-1] + 0.1*n[t]$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-017 presents the normalized blueprint; Teal TP-005 expresses IF/HOW/THEN; Brown BR-005 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-005); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-005.

Syntax expression for BP-ALG-006

```
module MJ.PBMEF.Blue.BP-ALG-006 {
  reference F-006;
  expression "F-006 = n_dot[t] = (2/3)*n_dot[t-1] + (1/3)*(n_bar[t]-n_bar[t-1])/dt";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-006: $n_dot[t] = (2/3)*n_dot[t-1] + (1/3)*(n_bar[t]-n_bar[t-1])/dt$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-019 presents the normalized blueprint; Teal TP-006 expresses IF/HOW/THEN; Brown BR-006 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-006); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-006.

Syntax expression for BP-ALG-007

```
module MJ.PBMEF.Blue.BP-ALG-007 {
  reference F-007;
  expression "F-007 = support>=3 of 5 AND 3 qualified windows";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-007: support>=3 of 5 AND 3 qualified windows

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-058 presents the normalized blueprint; Teal TP-007 expresses IF/HOW/THEN; Brown BR-007 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-007); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-007.

Syntax expression for BP-ALG-008

```
module MJ.PBMEF.Blue.BP-ALG-008 {
  reference F-008;
  expression "F-008 = g_i = 0.1*(2/3)^(i-1)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-008: $g_i = 0.1 \cdot (2/3)^{(i-1)}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-005 presents the normalized blueprint; Teal TP-008 expresses IF/HOW/THEN; Brown BR-008 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-008); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-008.

Syntax expression for BP-ALG-009

```
module MJ.PBMEF.Blue.BP-ALG-009 {
  reference F-009;
  expression "F-009 = Sigma_next = 0.9*Sigma_prev + 0.1*e*e^T";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-009: $\text{Sigma_next} = 0.9 \cdot \text{Sigma_prev} + 0.1 \cdot e \cdot e^T$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-055 presents the normalized blueprint; Teal TP-009 expresses IF/HOW/THEN; Brown BR-009 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-009); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-009.

Syntax expression for BP-ALG-010

```
module MJ.PBMEF.Blue.BP-ALG-010 {
  reference F-010;
  expression "F-010 = H = sqrt(lambda_max(Sigma))";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-010: $H = \sqrt{\lambda_{\max}(\Sigma)}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-054 presents the normalized blueprint; Teal TP-010 expresses IF/HOW/THEN; Brown BR-010 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-010); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-010.

Syntax expression for BP-ALG-011

```
module MJ.PBMEF.Blue.BP-ALG-011 {
  reference F-011;
  expression "F-011 = v_shadow = (H_next-H_prev)/dt";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-011: $v_shadow = (H_next - H_prev) / dt$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-004 presents the normalized blueprint; Teal TP-011 expresses IF/HOW/THEN; Brown BR-011 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-011); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-011.

Syntax expression for BP-ALG-012

```
module MJ.PBMEF.Blue.BP-ALG-012 {
  reference F-012, F-013;
  expression "F-012 = a0 = 48383829/100000000 ; F-013 = r0 = 51616171/100000000";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-012: a0 = 48383829/100000000; F-013: r0 = 51616171/100000000

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-077 presents the normalized blueprint; Teal TP-012 expresses IF/HOW/THEN; Brown BR-012 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-012, F-013); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-012.

Syntax expression for BP-ALG-013

```
module MJ.PBMEF.Blue.BP-ALG-013 {
  reference F-014;
  expression "F-014 = w_next = projection((2/3)w + (1/3)w_target)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-014: $w_next = projection((2/3)w + (1/3)w_target)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-066 presents the normalized blueprint; Teal TP-013 expresses IF/HOW/THEN; Brown BR-013 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-014); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-013.

Syntax expression for BP-ALG-014

```
module MJ.PBMEF.Blue.BP-ALG-014 {
  reference F-015;
  expression "F-015 = M_request = lambda_hat * bytes_per_request * latency_hat";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-015: $M_request = \lambda_hat * bytes_per_request * latency_hat$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-098 presents the normalized blueprint; Teal TP-014 expresses IF/HOW/THEN; Brown BR-014 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-015); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-014.

Syntax expression for BP-ALG-015

```
module MJ.PBMEF.Blue.BP-ALG-015 {
  reference F-016;
  expression "F-016 = M_pre = M_request + M_sync + M_residual + M_evidence + M_headroom";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-016: $M_pre = M_request + M_sync + M_residual + M_evidence + M_headroom$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-076 presents the normalized blueprint; Teal TP-015 expresses IF/HOW/THEN; Brown BR-015 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-016); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-015.

Syntax expression for BP-ALG-016

```
module MJ.PBMEF.Blue.BP-ALG-016 {
  reference F-017;
  expression "F-017 = QUIESCE -> SEAL -> CHECKPOINT -> DETACH -> RECLAIM";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-017: QUIESCE -> SEAL -> CHECKPOINT -> DETACH -> RECLAIM

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-088 presents the normalized blueprint; Teal TP-016 expresses IF/HOW/THEN; Brown BR-016 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-017); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-016.

Syntax expression for BP-ALG-017

```
module MJ.PBMEF.Blue.BP-ALG-017 {
  reference F-018;
  expression "F-018 = dQ/dt = I_intake - I_leak - I_release";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-018: $dQ/dt = I_{\text{intake}} - I_{\text{leak}} - I_{\text{release}}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-085 presents the normalized blueprint; Teal TP-017 expresses IF/HOW/THEN; Brown BR-017 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-018); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-017.

Syntax expression for BP-ALG-018

```
module MJ.PBMEF.Blue.BP-ALG-018 {
  reference F-019;
  expression "F-019 = E = 0.5*C*V^2";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-019: $E = 0.5 \cdot C \cdot V^2$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-055 presents the normalized blueprint; Teal TP-018 expresses IF/HOW/THEN; Brown BR-018 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-019); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-018.

Syntax expression for BP-ALG-019

```
module MJ.PBMEF.Blue.BP-ALG-019 {
  reference F-020;
  expression "F-020 = E = integral V(q)dq";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-020: $E = \text{integral } V(q)dq$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-033 presents the normalized blueprint; Teal TP-019 expresses IF/HOW/THEN; Brown BR-019 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-020); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-019.

Syntax expression for BP-ALG-020

```
module MJ.PBMEF.Blue.BP-ALG-020 {
  reference F-021;
  expression "F-021 = r_E = E_in - DeltaE_storage - E_delivered - E_containment - E_loss";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-021: $r_E = E_{in} - \Delta E_{storage} - E_{delivered} - E_{containment} - E_{loss}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-033 presents the normalized blueprint; Teal TP-020 expresses IF/HOW/THEN; Brown BR-020 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-021); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-020.

Syntax expression for BP-ALG-021

```
module MJ.PBMEF.Blue.BP-ALG-021 {
  reference F-022, F-023;
  expression "F-022 = r_M = M_initial + M_inputs - M_final - M_outputs ; F-023 = M_screen_in = M_captured + M_passed + M_retained + r_screen";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-022: $r_M = M_{\text{initial}} + M_{\text{inputs}} - M_{\text{final}} - M_{\text{outputs}}$; F-023: $M_{\text{screen_in}} = M_{\text{captured}} + M_{\text{passed}} + M_{\text{retained}} + r_{\text{screen}}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-087 presents the normalized blueprint; Teal TP-021 expresses IF/HOW/THEN; Brown BR-021 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-022, F-023); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-021.

Syntax expression for BP-ALG-022

```
module MJ.PBMEF.Blue.BP-ALG-022 {
  reference F-024;
  expression "F-024 = M_evap = integral integral J_m dA dt";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-024: $M_{\text{evap}} = \text{integral integral } J_m \text{ dA dt}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-076 presents the normalized blueprint; Teal TP-022 expresses IF/HOW/THEN; Brown BR-022 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-024); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-022.

Syntax expression for BP-ALG-023

```
module MJ.PBMEF.Blue.BP-ALG-023 {
reference F-025;
expression "F-025 = F_z = qE_z + F_b - mg - D_z - F_contact - F_adhesion";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-025: $F_z = qE_z + F_b - mg - D_z - F_{\text{contact}} - F_{\text{adhesion}}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-017 presents the normalized blueprint; Teal TP-023 expresses IF/HOW/THEN; Brown BR-023 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-025); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-023.

Syntax expression for BP-ALG-024

```
module MJ.PBMEF.Blue.BP-ALG-024 {
  reference F-026;
  expression "F-026 = p_P(x,y,z)=o_P+x*a_P+y*b_P+z*c_P";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-026: $p_P(x,y,z)=o_P+x*a_P+y*b_P+z*c_P$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-060 presents the normalized blueprint; Teal TP-024 expresses IF/HOW/THEN; Brown BR-024 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-026); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-024.

Syntax expression for BP-ALG-025

```
module MJ.PBMEF.Blue.BP-ALG-025 {
  reference F-027;
  expression "F-027 = N_U = max_k ceil(E_k/((1-h_k)*C_Uk))";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-027: $N_U = \max_k \text{ceil}(E_k / ((1-h_k) * C_{Uk}))$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-038 presents the normalized blueprint; Teal TP-025 expresses IF/HOW/THEN; Brown BR-025 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-027); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-025.

Syntax expression for BP-ALG-026

```
module MJ.PBMEF.Blue.BP-ALG-026 {
  reference F-028;
  expression "F-028 = r_B = B_alloc-B_active-B_released-B_retained-B_exported-B_compacted";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-028: $r_B = B_{\text{alloc}} - B_{\text{active}} - B_{\text{released}} - B_{\text{retained}} - B_{\text{exported}} - B_{\text{compacted}}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-029 presents the normalized blueprint; Teal TP-026 expresses IF/HOW/THEN; Brown BR-026 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-028); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-026.

Syntax expression for BP-ALG-027

```
module MJ.PBMEF.Blue.BP-ALG-027 {
  reference F-029;
  expression "F-029 = C_B = integral B_residual(t)dt";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-029: $C_B = \int B_{\text{residual}}(t)dt$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-013 presents the normalized blueprint; Teal TP-027 expresses IF/HOW/THEN; Brown BR-027 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-029); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-027.

Syntax expression for BP-ALG-028

```
module MJ.PBMEF.Blue.BP-ALG-028 {
  reference F-030;
  expression "F-030 = L_protected_H = O_near_H+R_emergency_H+R_tax_H+R_fees_H+U_cash_H";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-030: $L_{\text{protected_H}} = O_{\text{near_H}} + R_{\text{emergency_H}} + R_{\text{tax_H}} + R_{\text{fees_H}} + U_{\text{cash_H}}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-074 presents the normalized blueprint; Teal TP-028 expresses IF/HOW/THEN; Brown BR-028 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-030); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-028.

Syntax expression for BP-ALG-029

```
module MJ.PBMEF.Blue.BP-ALG-029 {
  reference F-031;
  expression "F-031 = S_investable_L = max(0,L_available_L-L_protected_H)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-031: $S_{investable_L} = \max(0, L_{available_L} - L_{protected_H})$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-059 presents the normalized blueprint; Teal TP-029 expresses IF/HOW/THEN; Brown BR-029 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-031); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-029.

Syntax expression for BP-ALG-030

```
module MJ.PBMEF.Blue.BP-ALG-030 {
  reference F-032;
  expression "F-032 = A_risk_max = min(S_investable_L,L_risk,L_product,L_regulatory,L_custody,L_concentration)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-032: $A_risk_max = \min(S_investable_L, L_risk, L_product, L_regulatory, L_custody, L_concentration)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-085 presents the normalized blueprint; Teal TP-030 expresses IF/HOW/THEN; Brown BR-030 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-032); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-030.

Syntax expression for BP-ALG-031

```
module MJ.PBMEF.Blue.BP-ALG-031 {
  reference F-033;
  expression "F-033 = C_i = abs(V_i_H)/V_portfolio_L";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-033: $C_i = \text{abs}(V_i_H)/V_{\text{portfolio}_L}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-033 presents the normalized blueprint; Teal TP-031 expresses IF/HOW/THEN; Brown BR-031 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-033); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-031.

Syntax expression for BP-ALG-032

```
module MJ.PBMEF.Blue.BP-ALG-032 {
  reference F-034;
  expression "F-034 = sigma_p^2 = w^T Sigma w";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-034: $\sigma_p^2 = w^T \Sigma w$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-030 presents the normalized blueprint; Teal TP-032 expresses IF/HOW/THEN; Brown BR-032 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-034); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-032.

Syntax expression for BP-ALG-033

```
module MJ.PBMEF.Blue.BP-ALG-033 {
  reference F-035;
  expression "F-035 = F_capital_H = fees+spread+slippage+tax+custody+compliance+latency";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-035: $F_capital_H = fees+spread+slippage+tax+custody+compliance+latency$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-079 presents the normalized blueprint; Teal TP-033 expresses IF/HOW/THEN; Brown BR-033 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-035); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-033.

Syntax expression for BP-ALG-034

```
module MJ.PBMEF.Blue.BP-ALG-034 {
  reference F-036;
  expression "F-036 = O_L = Return_L-F_capital_H-RiskPenalty_H-LiquididityPenalty_H-RegulatoryUncertainty_H";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-036: $O_L = Return_L-F_capital_H-RiskPenalty_H-LiquididityPenalty_H-RegulatoryUncertainty_H$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-016 presents the normalized blueprint; Teal TP-034 expresses IF/HOW/THEN; Brown BR-034 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-036); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-034.

Syntax expression for BP-ALG-035

```
module MJ.PBMEF.Blue.BP-ALG-035 {
  reference F-037;
  expression "F-037 = A = sum(alpha_i*P_i)-sum(beta_j*N_j)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-037: $A = \sum(\alpha_i * P_i) - \sum(\beta_j * N_j)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-018 presents the normalized blueprint; Teal TP-035 expresses IF/HOW/THEN; Brown BR-035 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-037); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-035.

Syntax expression for BP-ALG-036

```
module MJ.PBMEF.Blue.BP-ALG-036 {
  reference F-038;
  expression "F-038 = V_eligible_L=max(0,V_verified_L-L_known_H-U_valuation_H)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-038: $V_{eligible_L} = \max(0, V_{verified_L} - L_{known_H-U_valuation_H})$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-047 presents the normalized blueprint; Teal TP-036 expresses IF/HOW/THEN; Brown BR-036 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-038); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-036.

Syntax expression for BP-ALG-037

```
module MJ.PBMEF.Blue.BP-ALG-037 {
  reference F-039;
  expression "F-039 = N_max=floor(V_eligible_L/B_unit_H)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-039: $N_{max} = \text{floor}(V_{eligible_L} / B_{unit_H})$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-093 presents the normalized blueprint; Teal TP-037 expresses IF/HOW/THEN; Brown BR-037 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-039); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-037.

Syntax expression for BP-ALG-038

```
module MJ.PBMEF.Blue.BP-ALG-038 {
  reference F-040;
  expression "F-040 = x[t+1]=F(x[t],u[t],theta,e[t])+w[t]; y[t]=H(x[t])+v[t]";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-040: $x[t+1]=F(x[t],u[t],\theta,e[t])+w[t]; y[t]=H(x[t])+v[t]$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-057 presents the normalized blueprint; Teal TP-038 expresses IF/HOW/THEN; Brown BR-038 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-040); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-038.

Syntax expression for BP-ALG-039

```
module MJ.PBMEF.Blue.BP-ALG-039 {
  reference F-041;
  expression "F-041 = dim(LHS)=dim(RHS)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-041: $\text{dim(LHS)}=\text{dim(RHS)}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-054 presents the normalized blueprint; Teal TP-039 expresses IF/HOW/THEN; Brown BR-039 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-041); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-039.

Syntax expression for BP-ALG-040

```
module MJ.PBMEF.Blue.BP-ALG-040 {
  reference F-042;
  expression "F-042 = r_q=q_in-q_out-Deltaq-q_loss";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-042: $r_q=q_{in}-q_{out}-\Delta q-q_{loss}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-035 presents the normalized blueprint; Teal TP-040 expresses IF/HOW/THEN; Brown BR-040 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-042); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-040.

Syntax expression for BP-ALG-041

```
module MJ.PBMEF.Blue.BP-ALG-041 {
  reference F-043;
  expression "F-043 = V_P=(4/3)*pi*a*b*c";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-043: $V_P=(4/3)*\pi*a*b*c$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-094 presents the normalized blueprint; Teal TP-041 expresses IF/HOW/THEN; Brown BR-041 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-043); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-041.

Syntax expression for BP-ALG-042

```
module MJ.PBMEF.Blue.BP-ALG-042 {
  reference F-044;
  expression "F-044 = A_P~=4*pi*((a^p*b^p+a^p*c^p+b^p*c^p)/3)^(1/p), p=1.6075";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-044: $A_P \sim 4 \pi ((a^p b^p + a^p c^p + b^p c^p) / 3)^{(1/p)}$, $p = 1.6075$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-013 presents the normalized blueprint; Teal TP-042 expresses IF/HOW/THEN; Brown BR-042 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-044); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-042.

Syntax expression for BP-ALG-043

```
module MJ.PBMEF.Blue.BP-ALG-043 {
  reference F-045;
  expression "F-045 = chi_P=A_P/V_P";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-045: $\chi_P=A_P/V_P$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-023 presents the normalized blueprint; Teal TP-043 expresses IF/HOW/THEN; Brown BR-043 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-045); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-043.

Syntax expression for BP-ALG-044

```
module MJ.PBMEF.Blue.BP-ALG-044 {
  reference F-046;
  expression "F-046 = S_Pk=(1-h_Pk)*C_Pk-L_Pk";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-046: $S_Pk=(1-h_Pk)*C_Pk-L_Pk$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-073 presents the normalized blueprint; Teal TP-044 expresses IF/HOW/THEN; Brown BR-044 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-046); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-044.

Syntax expression for BP-ALG-045

```
module MJ.PBMEF.Blue.BP-ALG-045 {
  reference F-047;
  expression "F-047 = beta_P=max_k(L_Pk/((1-h_Pk)*C_Pk))";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-047: $\text{beta_P}=\text{max_k}(\text{L_Pk}/((1-\text{h_Pk})*\text{C_Pk}))$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-070 presents the normalized blueprint; Teal TP-045 expresses IF/HOW/THEN; Brown BR-045 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-047); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-045.

Syntax expression for BP-ALG-046

```
module MJ.PBMEF.Blue.BP-ALG-046 {
  reference F-048;
  expression "F-048 = N_P=max_k ceil(E_k/((1-h_Pk)*C_Pk))";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-048: $N_P = \max_k \text{ceil}(E_k / ((1 - h_{Pk}) * C_{Pk}))$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-058 presents the normalized blueprint; Teal TP-046 expresses IF/HOW/THEN; Brown BR-046 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-048); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-046.

Syntax expression for BP-ALG-047

```
module MJ.PBMEF.Blue.BP-ALG-047 {
  reference F-049;
  expression "F-049 = r_Pk=q_in-q_captured-q_transferred-q_retained-q_escaped-Deltaq_stored";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-049: $r_Pk=q_in-q_captured-q_transferred-q_retained-q_escaped-Deltaq_stored$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-043 presents the normalized blueprint; Teal TP-047 expresses IF/HOW/THEN; Brown BR-047 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-049); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-047.

Syntax expression for BP-ALG-048

```
module MJ.PBMEF.Blue.BP-ALG-048 {
  reference F-050;
  expression "F-050 = T_ij,k=min([S_jk]_,[Q_ik-Q_reserve_ik]_,C_route_k)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-050: $T_{ij,k} = \min([S_{jk}]_+, [Q_{ik} - Q_{reserve_ik}]_+, C_{route_k})$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-018 presents the normalized blueprint; Teal TP-048 expresses IF/HOW/THEN; Brown BR-048 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-050); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-048.

Syntax expression for BP-ALG-049

```
module MJ.PBMEF.Blue.BP-ALG-049 {
  reference F-051;
  expression "F-051 = G_P=[G_mass,G_energy,G_bytes,G_time,G_capital]";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-051: $G_P=[G_mass,G_energy,G_bytes,G_time,G_capital]$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-067 presents the normalized blueprint; Teal TP-049 expresses IF/HOW/THEN; Brown BR-049 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-051); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-049.

Syntax expression for BP-ALG-050

```
module MJ.PBMEF.Blue.BP-ALG-050 {
  reference F-052;
  expression "F-052 = lambda_ac=c_medium/f_control";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-052: $\lambda_{ac}=c_{medium}/f_{control}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-076 presents the normalized blueprint; Teal TP-050 expresses IF/HOW/THEN; Brown BR-050 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-052); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-050.

Syntax expression for BP-ALG-051

```
module MJ.PBMEF.Blue.BP-ALG-051 {
  reference F-053;
  expression "F-053 = I_ac=p_rms^2/(rho_medium*c_medium)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-053: $I_{ac}=p_{rms}^2/(rho_medium*c_medium)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-042 presents the normalized blueprint; Teal TP-051 expresses IF/HOW/THEN; Brown BR-051 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-053); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-051.

Syntax expression for BP-ALG-052

```
module MJ.PBMEF.Blue.BP-ALG-052 {
  reference F-054;
  expression "F-054 = E_ac=integral(I_ac*A_effective)dt";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-054: $E_{ac} = \int I_{ac} \cdot A_{effective} dt$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-054 presents the normalized blueprint; Teal TP-052 expresses IF/HOW/THEN; Brown BR-052 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-054); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-052.

Syntax expression for BP-ALG-053

```
module MJ.PBMEF.Blue.BP-ALG-053 {
  reference F-055;
  expression "F-055 = f_sample>=2*gamma*f_control AND f_control in calibrated_band AND SNR_f>=SNR_min";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-055: $f_sample \geq 2 * \gamma * f_control$ AND $f_control$ in calibrated_band AND $SNR_f \geq SNR_min$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-065 presents the normalized blueprint; Teal TP-053 expresses IF/HOW/THEN; Brown BR-053 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-055); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-053.

Syntax expression for BP-ALG-054

```
module MJ.PBMEF.Blue.BP-ALG-054 {
  reference F-056;
  expression "F-056 = SNR_f=P_signal(f)/P_noise(f)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-056: $SNR_f = P_signal(f) / P_noise(f)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-019 presents the normalized blueprint; Teal TP-054 expresses IF/HOW/THEN; Brown BR-054 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-056); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-054.

Syntax expression for BP-ALG-055

```
module MJ.PBMEF.Blue.BP-ALG-055 {
  reference F-057;
  expression "F-057 = H_P(f)=Y_P(f)/U_P(f)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-057: $H_P(f)=Y_P(f)/U_P(f)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-072 presents the normalized blueprint; Teal TP-055 expresses IF/HOW/THEN; Brown BR-055 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-057); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-055.

Syntax expression for BP-ALG-056

```
module MJ.PBMEF.Blue.BP-ALG-056 {
  reference F-058, F-059;
  expression "F-058 = f*=argmin_{f in F_safe}(J_breach+lambda_E*E_ac+lambda_T*T_transfer) ; F-059 =
  f_next=clip((2/3)f+(1/3)f_target,f-df_max,f+df_max,f_min,f_max)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-058: $f^*=argmin_{f \text{ in } F_safe}\{J_breach+lambda_E*E_ac+lambda_T*T_transfer\}$; F-059: $f_next=clip((2/3)f+(1/3)f_target,f-df_max,f+df_max,f_min,f_max)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-091 presents the normalized blueprint; Teal TP-056 expresses IF/HOW/THEN; Brown BR-056 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-058, F-059); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-056.

Syntax expression for BP-ALG-057

```
module MJ.PBMEF.Blue.BP-ALG-057 {
reference F-060;
expression "F-060 = r_registry=N_allocated+N_received-N_transferred-N_closed-N_active";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-060: $r_registry=N_allocated+N_received-N_transferred-N_closed-N_active$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-097 presents the normalized blueprint; Teal TP-057 expresses IF/HOW/THEN; Brown BR-057 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-060); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-057.

Syntax expression for BP-ALG-058

```
module MJ.PBMEF.Blue.BP-ALG-058 {
  reference F-061;
  expression "F-061 = C_ij=<value,unit,uncertainty,formula,revision,lineage,authority,evidence>";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-061: C_ij=<value,unit,uncertainty,formula,revision,lineage,authority,evidence>

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-089 presents the normalized blueprint; Teal TP-058 expresses IF/HOW/THEN; Brown BR-058 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-061); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-058.

Syntax expression for BP-ALG-059

```
module MJ.PBMEF.Blue.BP-ALG-059 {
  reference F-062;
  expression "F-062 = A(U)=Reachable_G(U)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-062: $A(U)=Reachable_G(U)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-043 presents the normalized blueprint; Teal TP-059 expresses IF/HOW/THEN; Brown BR-059 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-062); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-059.

Syntax expression for BP-ALG-060

```
module MJ.PBMEF.Blue.BP-ALG-060 {
  reference F-063;
  expression "F-063 = C_recalc=sum_{v in A}c_v+sum_{e in E(A)}t_e";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-063: $C_recalc = \sum_{v \in A} c_v + \sum_{e \in E(A)} t_e$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-091 presents the normalized blueprint; Teal TP-060 expresses IF/HOW/THEN; Brown BR-060 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-063); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-060.

Syntax expression for BP-ALG-061

```
module MJ.PBMEF.Blue.BP-ALG-061 {
  reference F-064;
  expression "F-064 = lambda_total=N_users*lambda_user";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-064: lambda_total=N_users*lambda_user

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-015 presents the normalized blueprint; Teal TP-061 expresses IF/HOW/THEN; Brown BR-061 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-064); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-061.

Syntax expression for BP-ALG-062

```
module MJ.PBMEF.Blue.BP-ALG-062 {
  reference F-065;
  expression "F-065 = rho=lambda_total/(W*mu_worker)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-065: rho=lambda_total/(W*mu_worker)

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-028 presents the normalized blueprint; Teal TP-062 expresses IF/HOW/THEN; Brown BR-062 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-065); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-062.

Syntax expression for BP-ALG-063

```
module MJ.PBMEF.Blue.BP-ALG-063 {
  reference F-066;
  expression "F-066 = W_min=ceil(lambda_total/(rho_max*mu_worker));";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-066: $W_{min} = \text{ceil}(\lambda_{total} / (\rho_{max} * \mu_{worker}))$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-065 presents the normalized blueprint; Teal TP-063 expresses IF/HOW/THEN; Brown BR-063 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-066); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-063.

Syntax expression for BP-ALG-064

```
module MJ.PBMEF.Blue.BP-ALG-064 {
  reference F-067;
  expression "F-067 = T_recalc=T_parse+T_dependency+max_p(T_partition,p)+T_commit";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-067: $T_recalc=T_parse+T_dependency+max_p(T_partition,p)+T_commit$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-034 presents the normalized blueprint; Teal TP-064 expresses IF/HOW/THEN; Brown BR-064 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-067); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-064.

Syntax expression for BP-ALG-065

```
module MJ.PBMEF.Blue.BP-ALG-065 {
  reference F-068;
  expression "F-068 = C_data=B_store*c_store+B_transfer*c_transfer+t_cpu*c_cpu";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-068: $C_data=B_store*c_store+B_transfer*c_transfer+t_cpu*c_cpu$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-019 presents the normalized blueprint; Teal TP-065 expresses IF/HOW/THEN; Brown BR-065 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-068); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-065.

Syntax expression for BP-ALG-066

```
module MJ.PBMEF.Blue.BP-ALG-066 {
  reference F-069;
  expression "F-069 = J_partition=alpha*max_p(L_p)+beta*sum_cut(w_e)+gamma*R_breach+delta*R_stale";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-069: $J_partition = \alpha * \max_p(L_p) + \beta * \text{sum_cut}(w_e) + \gamma * R_breach + \delta * R_stale$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-081 presents the normalized blueprint; Teal TP-066 expresses IF/HOW/THEN; Brown BR-066 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-069); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-066.

Syntax expression for BP-ALG-067

```
module MJ.PBMEF.Blue.BP-ALG-067 {
  reference F-070;
  expression "F-070 = H_rev=SHA256(parent_root||canonical_deltas||policy_epoch)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-070: H_rev=SHA256(parent_root||canonical_deltas||policy_epoch)

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-049 presents the normalized blueprint; Teal TP-067 expresses IF/HOW/THEN; Brown BR-067 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-070); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-067.

Syntax expression for BP-ALG-068

```
module MJ.PBMEF.Blue.BP-ALG-068 {
  reference F-071;
  expression "F-071 = AuditPass=AND(required_fields_present,receipts_valid)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-071: AuditPass=AND(required_fields_present,receipts_valid)

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-069 presents the normalized blueprint; Teal TP-068 expresses IF/HOW/THEN; Brown BR-068 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-071); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-068.

Syntax expression for BP-ALG-069

```
module MJ.PBMEF.Blue.BP-ALG-069 {
  reference F-072;
  expression "F-072 = spectral_radius(J_F)<1";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-072: spectral_radius(J_F)<1

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-040 presents the normalized blueprint; Teal TP-069 expresses IF/HOW/THEN; Brown BR-069 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-072); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-069.

Syntax expression for BP-ALG-070

```
module MJ.PBMEF.Blue.BP-ALG-070 {
  reference F-073;
  expression "F-073 = base_revision=head_revision OR valid_merge_receipt";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-073: base_revision=head_revision OR valid_merge_receipt

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-066 presents the normalized blueprint; Teal TP-070 expresses IF/HOW/THEN; Brown BR-070 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-073); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-070.

Syntax expression for BP-ALG-071

```
module MJ.PBMEF.Blue.BP-ALG-071 {
  reference F-074;
  expression "F-074 = Age=now-observed_at";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-074: Age=now-observed_at

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-063 presents the normalized blueprint; Teal TP-071 expresses IF/HOW/THEN; Brown BR-071 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-074); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-071.

Syntax expression for BP-ALG-072

```
module MJ.PBMEF.Blue.BP-ALG-072 {
  reference F-075;
  expression "F-075 = Age<=freshness_limit";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-075: Age<=freshness_limit

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-046 presents the normalized blueprint; Teal TP-072 expresses IF/HOW/THEN; Brown BR-072 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-075); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-072.

Syntax expression for BP-ALG-073

```
module MJ.PBMEF.Blue.BP-ALG-073 {
  reference F-076;
  expression "F-076 = r_xy=Cov(X,Y)/(sigma_x*sigma_y)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-076: $r_{xy} = \text{Cov}(X,Y) / (\sigma_x \sigma_y)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-011 presents the normalized blueprint; Teal TP-073 expresses IF/HOW/THEN; Brown BR-073 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-076); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-073.

Syntax expression for BP-ALG-074

```
module MJ.PBMEF.Blue.BP-ALG-074 {
  reference F-077;
  expression "F-077 = UnifiedMetric=SameMeaning AND SameUnit AND SamePeriod AND SameAuthority AND SameLineage";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-077: UnifiedMetric=SameMeaning AND SameUnit AND SamePeriod AND SameAuthority AND SameLineage

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-006 presents the normalized blueprint; Teal TP-074 expresses IF/HOW/THEN; Brown BR-074 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-077); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-074.

Syntax expression for BP-ALG-075

```
module MJ.PBMEF.Blue.BP-ALG-075 {
  reference F-078;
  expression "F-078 = r_formula=y_reported-y_recomputed";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-078: r_formula=y_reported-y_recomputed

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-006 presents the normalized blueprint; Teal TP-075 expresses IF/HOW/THEN; Brown BR-075 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-078); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-075.

Syntax expression for BP-ALG-076

```
module MJ.PBMEF.Blue.BP-ALG-076 {
  reference F-079;
  expression "F-079 = IntakePass=process_defined AND users_known AND workload_known AND SLA_known AND authority_known";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-079: IntakePass=process_defined AND users_known AND workload_known AND SLA_known AND authority_known

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-088 presents the normalized blueprint; Teal TP-076 expresses IF/HOW/THEN; Brown BR-076 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-079); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-076.

Syntax expression for BP-ALG-077

```
module MJ.PBMEF.Blue.BP-ALG-077 {
reference F-080;
expression "F-080 = lambda_total<=rho_max*sum_p(mu_p) AND memory_required<=memory_available AND cross_partition_edges<=cut_limit";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-080: $\lambda_{total} \leq \rho_{max} \cdot \sum_p(\mu_p)$ AND $memory_required \leq memory_available$ AND $cross_partition_edges \leq cut_limit$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-097 presents the normalized blueprint; Teal TP-077 expresses IF/HOW/THEN; Brown BR-077 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-080); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-077.

Syntax expression for BP-ALG-078

```
module MJ.PBMEF.Blue.BP-ALG-078 {
  reference F-081;
  expression "F-081 = H_R=h_noise,1";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-081: H_R=h_noise,1

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-048 presents the normalized blueprint; Teal TP-078 expresses IF/HOW/THEN; Brown BR-078 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-081); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-078.

Syntax expression for BP-ALG-079

```
module MJ.PBMEF.Blue.BP-ALG-079 {
  reference F-082;
  expression "F-082 = rho_g=P*M/(Z*R*T)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-082: $\rho_g=P*M/(Z*R*T)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-075 presents the normalized blueprint; Teal TP-079 expresses IF/HOW/THEN; Brown BR-079 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-082); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-079.

Syntax expression for BP-ALG-080

```
module MJ.PBMEF.Blue.BP-ALG-080 {
  reference F-083;
  expression "F-083 = V_24=m_24/rho_ref";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-083: $V_{24}=m_{24}/\rho_{ref}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-092 presents the normalized blueprint; Teal TP-080 expresses IF/HOW/THEN; Brown BR-080 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-083); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-080.

Syntax expression for BP-ALG-081

```
module MJ.PBMEF.Blue.BP-ALG-081 {
  reference F-084;
  expression "F-084 = D_R=V_24/(W_R*H_R)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-084: $D_R=V_{24}/(W_R \cdot H_R)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-069 presents the normalized blueprint; Teal TP-081 expresses IF/HOW/THEN; Brown BR-081 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-084); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-081.

Syntax expression for BP-ALG-082

```
module MJ.PBMEF.Blue.BP-ALG-082 {
reference F-085, F-086;
expression "F-085 = H_C=k_H*H_R; k_H,std=2 ; F-086 = W_T=k_W*W_B; k_W,std=2";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-085: $H_C=k_H*H_R$; $k_H,std=2$; F-086: $W_T=k_W*W_B$; $k_W,std=2$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-003 presents the normalized blueprint; Teal TP-082 expresses IF/HOW/THEN; Brown BR-082 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-085, F-086); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-082.

Syntax expression for BP-ALG-083

```
module MJ.PBMEF.Blue.BP-ALG-083 {
  reference F-087;
  expression "F-087 = A(z)=A_B*(1+(k_W-1)*z/H_C)^2";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-087: $A(z)=A_B*(1+(k_W-1)*z/H_C)^2$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-027 presents the normalized blueprint; Teal TP-083 expresses IF/HOW/THEN; Brown BR-083 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-087); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-083.

Syntax expression for BP-ALG-084

```
module MJ.PBMEF.Blue.BP-ALG-084 {
  reference F-088;
  expression "F-088 = V_C(z)=A_B*(z+(k_W-1)z^2/H_C+(k_W-1)^2*z^3/(3H_C^2))";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-088: $V_C(z)=A_B(z+(k_W-1)z^2/H_C+(k_W-1)^2z^3/(3H_C^2))$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-090 presents the normalized blueprint; Teal TP-084 expresses IF/HOW/THEN; Brown BR-084 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-088); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-084.

Syntax expression for BP-ALG-085

```
module MJ.PBMEF.Blue.BP-ALG-085 {
  reference F-089;
  expression "F-089 = V_C=H_C/3*(A_B+sqrt(A_B*A_T)+A_T)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-089: $V_C=H_C/3*(A_B+sqrt(A_B*A_T)+A_T)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-018 presents the normalized blueprint; Teal TP-085 expresses IF/HOW/THEN; Brown BR-085 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-089); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-085.

Syntax expression for BP-ALG-086

```
module MJ.PBMEF.Blue.BP-ALG-086 {
  reference F-090;
  expression "F-090 = z_P0=H_R+H_C; standard=3H_R";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-090: z_P0=H_R+H_C; standard=3H_R

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-078 presents the normalized blueprint; Teal TP-086 expresses IF/HOW/THEN; Brown BR-086 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-090); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-086.

Syntax expression for BP-ALG-087

```
module MJ.PBMEF.Blue.BP-ALG-087 {
  reference F-091;
  expression "F-091 = m_g=rho_g*V_C(h_g)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-091: $m_g = \rho_g \cdot V_C(h_g)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-080 presents the normalized blueprint; Teal TP-087 expresses IF/HOW/THEN; Brown BR-087 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-091); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-087.

Syntax expression for BP-ALG-088

```
module MJ.PBMEF.Blue.BP-ALG-088 {
  reference F-092;
  expression "F-092 = m_dot=rho*A(h_g)*h_dot_g+V_C(h_g)*rho_dot";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-092: $m_{\dot{}}=\rho\cdot A(h_g)\cdot h_{\dot{}}_g+V_C(h_g)\cdot \rho_{\dot{}}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-005 presents the normalized blueprint; Teal TP-088 expresses IF/HOW/THEN; Brown BR-088 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-092); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-088.

Syntax expression for BP-ALG-089

```
module MJ.PBMEF.Blue.BP-ALG-089 {
  reference F-093;
  expression "F-093 = h_hat=clip(h+v*dt+0.5*a*dt^2,0,H_C)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-093: $h_hat=clip(h+v*dt+0.5*a*dt^2,0,H_C)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-045 presents the normalized blueprint; Teal TP-089 expresses IF/HOW/THEN; Brown BR-089 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-093); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-089.

Syntax expression for BP-ALG-090

```
module MJ.PBMEF.Blue.BP-ALG-090 {
  reference F-094;
  expression "F-094 = m_eff*a_z=F_b+F_p+F_act-F_d-F_wall-F_mix";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-094: $m_{eff}a_z=F_b+F_p+F_{act}-F_d-F_{wall}-F_{mix}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-059 presents the normalized blueprint; Teal TP-090 expresses IF/HOW/THEN; Brown BR-090 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-094); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-090.

Syntax expression for BP-ALG-091

```
module MJ.PBMEF.Blue.BP-ALG-091 {
  reference F-095, F-096;
  expression "F-095 = F_b=(rho_ambient-rho_g)*g*V_g ; F-096 = F_d=0.5*C_d*rho_ambient*A_projected*u*abs(u)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-095: $F_b=(\rho_{\text{ambient}}-\rho_g)*g*V_g$; F-096: $F_d=0.5*C_d*\rho_{\text{ambient}}*A_{\text{projected}}*u*abs(u)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-098 presents the normalized blueprint; Teal TP-091 expresses IF/HOW/THEN; Brown BR-091 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-095, F-096); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-091.

Syntax expression for BP-ALG-092

```
module MJ.PBMEF.Blue.BP-ALG-092 {
  reference F-097;
  expression "F-097 = Q=A(z)*u(z)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-097: $Q=A(z)*u(z)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-091 presents the normalized blueprint; Teal TP-092 expresses IF/HOW/THEN; Brown BR-092 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-097); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-092.

Syntax expression for BP-ALG-093

```
module MJ.PBMEF.Blue.BP-ALG-093 {
  reference F-098;
  expression "F-098 = C_res,k=gamma_k*C_rect,k; 0<=gamma_k<1";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-098: $C_{res,k} = \gamma_k C_{rect,k}; 0 \leq \gamma_k < 1$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-003 presents the normalized blueprint; Teal TP-093 expresses IF/HOW/THEN; Brown BR-093 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-098); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-093.

Syntax expression for BP-ALG-094

```
module MJ.PBMEF.Blue.BP-ALG-094 {
  reference F-099;
  expression "F-099 = q_cone=q_down+q_EVE+q_res+q_escape+r";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-099: $q_{cone}=q_{down}+q_{EVE}+q_{res}+q_{escape}+r$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-034 presents the normalized blueprint; Teal TP-094 expresses IF/HOW/THEN; Brown BR-094 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-099); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-094.

Syntax expression for BP-ALG-095

```
module MJ.PBMEF.Blue.BP-ALG-095 {
  reference F-100;
  expression "F-100 = Valid(theta)=typed&positive&recomputed&conserved&bounded&evidenced&MK&JO&MJ";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-100: Valid(theta)=typed&positive&recomputed&conserved&bounded&evidenced&MK&JO&MJ

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-066 presents the normalized blueprint; Teal TP-095 expresses IF/HOW/THEN; Brown BR-095 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-100); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-095.

Syntax expression for BP-ALG-096

```
module MJ.PBMEF.Blue.BP-ALG-096 {
  reference F-101;
  expression "F-101 = z_out=H_P-delta_z; 0<delta_z<H_P";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-101: $z_{out}=H_P-\delta z; 0<\delta z<H_P$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-090 presents the normalized blueprint; Teal TP-096 expresses IF/HOW/THEN; Brown BR-096 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-101); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-096.

Syntax expression for BP-ALG-097

```
module MJ.PBMEF.Blue.BP-ALG-097 {
  reference F-102;
  expression "F-102 = R_P,sim=R_top+c_P; c_P>=0";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-102: $R_P,sim=R_{top}+c_P; c_P \geq 0$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-015 presents the normalized blueprint; Teal TP-097 expresses IF/HOW/THEN; Brown BR-097 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-102); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-097.

Syntax expression for BP-ALG-098

```
module MJ.PBMEF.Blue.BP-ALG-098 {
  reference F-103;
  expression "F-103 = d_out=cos(alpha_out)t+sin(alpha_out)n_out; 0<alpha_out<pi/2";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-103: $d_{out} = \cos(\alpha_{out})t + \sin(\alpha_{out})n_{out}; 0 < \alpha_{out} < \pi/2$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-025 presents the normalized blueprint; Teal TP-098 expresses IF/HOW/THEN; Brown BR-098 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-103); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-098.

Syntax expression for BP-ALG-099

```
module MJ.PBMEF.Blue.BP-ALG-099 {
  reference F-104;
  expression "F-104 = ||gamma(s)-c_P||^2=R_P^2+2R_P*s*sin(alpha_out)+s^2";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-104: $||\gamma(s)-c_P||^2=R_P^2+2R_P*s*\sin(\alpha_{out})+s^2$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-083 presents the normalized blueprint; Teal TP-099 expresses IF/HOW/THEN; Brown BR-099 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-104); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-099.

Syntax expression for BP-ALG-100

```
module MJ.PBMEF.Blue.BP-ALG-100 {
  reference F-105;
  expression "F-105 = dV_L/dt=Q_out-Q_return-Q_drain-Q_escape";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-105: $dV_L/dt=Q_{out}-Q_{return}-Q_{drain}-Q_{escape}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-034 presents the normalized blueprint; Teal TP-100 expresses IF/HOW/THEN; Brown BR-100 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-105); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-100.

Syntax expression for BP-ALG-101

```
module MJ.PBMEF.Blue.BP-ALG-101 {
  reference F-106;
  expression "F-106 = dq_L,k/dt=q_out,k-q_return,k-q_drain,k-q_escape,k";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-106: $dq_L,k/dt=q_{out,k}-q_{return,k}-q_{drain,k}-q_{escape,k}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-044 presents the normalized blueprint; Teal TP-101 expresses IF/HOW/THEN; Brown BR-101 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-106); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-101.

Syntax expression for BP-ALG-102

```
module MJ.PBMEF.Blue.BP-ALG-102 {
  reference F-107;
  expression "F-107 = V_L, req=T_hold*max(0,Q_out,H-Q_return,L)+V_residual,H";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-107: $V_L, req=T_hold*max(0,Q_out,H-Q_return,L)+V_residual,H$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-029 presents the normalized blueprint; Teal TP-102 expresses IF/HOW/THEN; Brown BR-102 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-107); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-102.

Syntax expression for BP-ALG-103

```
module MJ.PBMEF.Blue.BP-ALG-103 {
  reference F-108;
  expression "F-108 = A_out>=Q_design/u_out,max";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-108: $A_{out} \geq Q_{design}/u_{out,max}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-084 presents the normalized blueprint; Teal TP-103 expresses IF/HOW/THEN; Brown BR-103 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-108); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-103.

Syntax expression for BP-ALG-104

```
module MJ.PBMEF.Blue.BP-ALG-104 {
  reference F-109;
  expression "F-109 = r_out>=sqrt(A_out/pi)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-109: $r_{out} \geq \sqrt{A_{out}/\pi}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-006 presents the normalized blueprint; Teal TP-104 expresses IF/HOW/THEN; Brown BR-104 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-109); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-104.

Syntax expression for BP-ALG-105

```
module MJ.PBMEF.Blue.BP-ALG-105 {
  reference F-110;
  expression "F-110 = J_drop=m_drop*v_drop<=J_drop,max";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-110: $J_drop=m_drop*v_drop<=J_drop,max$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-022 presents the normalized blueprint; Teal TP-105 expresses IF/HOW/THEN; Brown BR-105 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-110); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-105.

Syntax expression for BP-ALG-106

```
module MJ.PBMEF.Blue.BP-ALG-106 {
  reference F-111;
  expression "F-111 = E_wave~0.5*rho*g*a_wave^2*A_surface";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-111: $E_wave \sim 0.5 \cdot \rho \cdot g \cdot a_wave^2 \cdot A_surface$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-075 presents the normalized blueprint; Teal TP-106 expresses IF/HOW/THEN; Brown BR-106 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-111); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-106.

Syntax expression for BP-ALG-107

```
module MJ.PBMEF.Blue.BP-ALG-107 {
  reference F-112;
  expression "F-112 = Fr=U/sqrt(gL)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-112: Fr=U/sqrt(gL)

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-085 presents the normalized blueprint; Teal TP-107 expresses IF/HOW/THEN; Brown BR-107 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-112); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-107.

Syntax expression for BP-ALG-108

```
module MJ.PBMEF.Blue.BP-ALG-108 {
  reference F-113;
  expression "F-113 = A_in=pi*r_in^2";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-113: $A_{in}=\pi*r_{in}^2$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-038 presents the normalized blueprint; Teal TP-108 expresses IF/HOW/THEN; Brown BR-108 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-113); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-108.

Syntax expression for BP-ALG-109

```
module MJ.PBMEF.Blue.BP-ALG-109 {
reference F-114;
expression "F-114 = u_r=-U*sin(beta_in); u_theta=sigma*U*cos(beta_in)";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-114: $u_r=-U\sin(\beta_{in}); u_{\theta}=\sigma U\cos(\beta_{in})$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-017 presents the normalized blueprint; Teal TP-109 expresses IF/HOW/THEN; Brown BR-109 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-114); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-109.

Syntax expression for BP-ALG-110

```
module MJ.PBMEF.Blue.BP-ALG-110 {
  reference F-115;
  expression "F-115 = chi_spiral=|u_theta|/max(|u_r|,epsilon)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-115: $\text{chi_spiral} = |u_theta| / \max(|u_r|, \epsilon)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-016 presents the normalized blueprint; Teal TP-110 expresses IF/HOW/THEN; Brown BR-110 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-115); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-110.

Syntax expression for BP-ALG-111

```
module MJ.PBMEF.Blue.BP-ALG-111 {
  reference F-116;
  expression "F-116 = Q_return=A_in*u_n";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-116: Q_return=A_in*u_n

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-018 presents the normalized blueprint; Teal TP-111 expresses IF/HOW/THEN; Brown BR-111 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-116); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-111.

Syntax expression for BP-ALG-112

```
module MJ.PBMEF.Blue.BP-ALG-112 {
  reference F-117;
  expression "F-117 = v_rel=v_B-v_A-Omega_A x r";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-117: $v_{rel}=v_B-v_A-\Omega_A \times r$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-073 presents the normalized blueprint; Teal TP-112 expresses IF/HOW/THEN; Brown BR-112 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-117); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-112.

Syntax expression for BP-ALG-113

```
module MJ.PBMEF.Blue.BP-ALG-113 {
  reference F-118;
  expression "F-118 = P_diss=max(0,-F_f dot v_rel)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-118: $P_{diss} = \max(0, -F_f \cdot v_{rel})$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-022 presents the normalized blueprint; Teal TP-113 expresses IF/HOW/THEN; Brown BR-113 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-118); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-113.

Syntax expression for BP-ALG-114

```
module MJ.PBMEF.Blue.BP-ALG-114 {
  reference F-119;
  expression "F-119 = E_rec<=eta_rec*integral(P_diss dt); 0<=eta_rec<=1";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-119: $E_{rec} \leq \eta_{rec} \cdot \int P_{diss} dt$; $0 \leq \eta_{rec} \leq 1$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-064 presents the normalized blueprint; Teal TP-114 expresses IF/HOW/THEN; Brown BR-114 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-119); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-114.

Syntax expression for BP-ALG-115

```
module MJ.PBMEF.Blue.BP-ALG-115 {
  reference F-120;
  expression "F-120 = Admit=AngelPass&M3Lease&MKPass&JOPass&MermaidReconciled&MJPermit";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-120: Admit=AngelPass&M3Lease&MKPass&JOPass&MermaidReconciled&MJPermit

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-065 presents the normalized blueprint; Teal TP-115 expresses IF/HOW/THEN; Brown BR-115 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-120); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-115.

Syntax expression for BP-ALG-116

```
module MJ.PBMEF.Blue.BP-ALG-116 {
  reference F-121;
  expression "F-121 = chi_i=|u_theta|/max(|u_r|,epsilon_u)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-121: $chi_i=|u_{theta}|/max(|u_r|,epsilon_u)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-070 presents the normalized blueprint; Teal TP-116 expresses IF/HOW/THEN; Brown BR-116 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-121); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-116.

Syntax expression for BP-ALG-117

```
module MJ.PBMEF.Blue.BP-ALG-117 {
  reference F-122;
  expression "F-122 = cycle_reentry=cycle_exit+1";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-122: cycle_reentry=cycle_exit+1

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-045 presents the normalized blueprint; Teal TP-117 expresses IF/HOW/THEN; Brown BR-117 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-122); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-117.

Syntax expression for BP-ALG-118

```
module MJ.PBMEF.Blue.BP-ALG-118 {
  reference F-123;
  expression "F-123 = r_Pi=x_B-T_A_to_B*x_A";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-123: $r_{Pi} = x_{B-T_A_to_B} \cdot x_A$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-003 presents the normalized blueprint; Teal TP-118 expresses IF/HOW/THEN; Brown BR-118 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-123); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-118.

Syntax expression for BP-ALG-119

```
module MJ.PBMEF.Blue.BP-ALG-119 {
  reference F-124;
  expression "F-124 = E_diss=int(Delta_p(t)*Q(t)dt)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-124: $E_{diss} = \int \Delta p(t) \cdot Q(t) dt$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-086 presents the normalized blueprint; Teal TP-119 expresses IF/HOW/THEN; Brown BR-119 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-124); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-119.

Syntax expression for BP-ALG-120

```
module MJ.PBMEF.Blue.BP-ALG-120 {
  reference F-125;
  expression "F-125 = C_info=int(B_residual(t)dt)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-125: C_info=int(B_residual(t)dt)

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-046 presents the normalized blueprint; Teal TP-120 expresses IF/HOW/THEN; Brown BR-120 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-125); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-120.

Syntax expression for BP-ALG-121

```
module MJ.PBMEF.Blue.BP-ALG-121 {
  reference F-126;
  expression "F-126 = omega_tilde=(2/3)*omega+(1/3)*omega_star";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-126: $\omega_{\text{tilde}}=(2/3)\omega+(1/3)\omega_{\text{star}}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-051 presents the normalized blueprint; Teal TP-121 expresses IF/HOW/THEN; Brown BR-121 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-126); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-121.

Syntax expression for BP-ALG-122

```
module MJ.PBMEF.Blue.BP-ALG-122 {
  reference F-127;
  expression "F-127 = |omega_next-omega|<=0.1*omega_scale";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-127: |omega_next-omega|<=0.1*omega_scale

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-025 presents the normalized blueprint; Teal TP-122 expresses IF/HOW/THEN; Brown BR-122 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-127); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-122.

Syntax expression for BP-ALG-123

```
module MJ.PBMEF.Blue.BP-ALG-123 {
  reference F-128;
  expression "F-128 = q_exit=q_lake_in+q_EVE+q_escape+r_o";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-128: $q_{\text{exit}}=q_{\text{lake_in}}+q_{\text{EVE}}+q_{\text{escape}}+r_o$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-080 presents the normalized blueprint; Teal TP-123 expresses IF/HOW/THEN; Brown BR-123 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-128); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-123.

Syntax expression for BP-ALG-124

```
module MJ.PBMEF.Blue.BP-ALG-124 {
  reference F-129;
  expression "F-129 = Delta_x=[Delta_m,Delta_w,Delta_v,Delta_rho,Delta_T,Delta_p,Delta_h]";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-129: Delta_x=[Delta_m,Delta_w,Delta_v,Delta_rho,Delta_T,Delta_p,Delta_h]

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-029 presents the normalized blueprint; Teal TP-124 expresses IF/HOW/THEN; Brown BR-124 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-129); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-124.

Syntax expression for BP-ALG-125

```
module MJ.PBMEF.Blue.BP-ALG-125 {
  reference F-130;
  expression "F-130 = Admit_re=types&units&new_cycle&capacity&wave&MK&JO&Angels&M3&MJ";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-130: Admit_re=types&units&new_cycle&capacity&wave&MK&JO&Angels&M3&MJ

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-053 presents the normalized blueprint; Teal TP-125 expresses IF/HOW/THEN; Brown BR-125 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-130); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-125.

Syntax expression for BP-ALG-126

```
module MJ.PBMEF.Blue.BP-ALG-126 {
  reference F-131;
  expression "F-131 = min_s(||gamma_out(s)-c_P||-R_P,sim) >= c_min+U_path";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-131: $\min_s(\| \gamma_{out}(s) - c_P \| - R_P, sim) \geq c_{min} + U_{path}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-028 presents the normalized blueprint; Teal TP-126 expresses IF/HOW/THEN; Brown BR-126 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-131); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-126.

Syntax expression for BP-ALG-127

```
module MJ.PBMEF.Blue.BP-ALG-127 {
  reference F-132;
  expression "F-132 = abs(wrap(alpha_out-alpha_gap)) <= Delta_alpha_gap/2";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-132: $\text{abs}(\text{wrap}(\alpha_{\text{out}}-\alpha_{\text{gap}})) \leq \Delta_{\alpha_{\text{gap}}}/2$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-075 presents the normalized blueprint; Teal TP-127 expresses IF/HOW/THEN; Brown BR-127 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-132); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-127.

Syntax expression for BP-ALG-128

```
module MJ.PBMEF.Blue.BP-ALG-128 {
  reference F-133;
  expression "F-133 = Delta_p_available=p_containment-p_lake+rho*g*(z_out-z_lake)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-133: $\Delta p_{\text{available}} = p_{\text{containment}} - p_{\text{lake}} + \rho \cdot g \cdot (z_{\text{out}} - z_{\text{lake}})$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-019 presents the normalized blueprint; Teal TP-128 expresses IF/HOW/THEN; Brown BR-128 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-133); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-128.

Syntax expression for BP-ALG-129

```
module MJ.PBMEF.Blue.BP-ALG-129 {
  reference F-134;
  expression "F-134 = Delta_p_margin=Delta_p_available-Delta_p_loss-U_Delta_p";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-134: $\Delta p_{margin} = \Delta p_{available} - \Delta p_{loss} - U \Delta p$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-072 presents the normalized blueprint; Teal TP-129 expresses IF/HOW/THEN; Brown BR-129 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-134); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-129.

Syntax expression for BP-ALG-130

```
module MJ.PBMEF.Blue.BP-ALG-130 {
reference F-135, F-136;
expression "F-135 = alpha_min=atan(max(0,Delta_p_loss+U_Delta_p-(p_containment-p_lake))/(rho*g*L_path)) ; F-136 =
alpha_selected=alpha_min+delta_alpha; delta_alpha>0";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY
F-135: alpha_min=atan(max(0,Delta_p_loss+U_Delta_p-(p_containment-p_lake))/(rho*g*L_path)); F-136: alpha_selected=alpha_min+delta_alpha; delta_alpha>0

CODE -> MATH -> IMAGE -> PROCESS FLOW
Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-082 presents the normalized blueprint; Teal TP-130 expresses IF/HOW/THEN; Brown BR-130 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES
FORMULA-REGISTER.json (F-135, F-136); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-130.

Syntax expression for BP-ALG-131

```
module MJ.PBMEF.Blue.BP-ALG-131 {
  reference F-137;
  expression "F-137 = Q_ref=C_d*A_out*sqrt(2*max(0,Delta_p_available-Delta_p_loss)/rho)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-137: $Q_{ref}=C_dA_{out}\sqrt{2\max(0,\Delta p_{available}-\Delta p_{loss})/\rho}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-096 presents the normalized blueprint; Teal TP-131 expresses IF/HOW/THEN; Brown BR-131 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-137); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-131.

Syntax expression for BP-ALG-132

```
module MJ.PBMEF.Blue.BP-ALG-132 {
  reference F-138;
  expression "F-138 = U_out=Q_ref/A_out";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-138: $U_{out}=Q_{ref}/A_{out}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-025 presents the normalized blueprint; Teal TP-132 expresses IF/HOW/THEN; Brown BR-132 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-138); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-132.

Syntax expression for BP-ALG-133

```
module MJ.PBMEF.Blue.BP-ALG-133 {
  reference F-139;
  expression "F-139 = We=rho*U^2*L/sigma";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-139: $We=\rho \cdot U^2 \cdot L/\sigma$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-061 presents the normalized blueprint; Teal TP-133 expresses IF/HOW/THEN; Brown BR-133 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-139); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-133.

Syntax expression for BP-ALG-134

```
module MJ.PBMEF.Blue.BP-ALG-134 {
  reference F-140;
  expression "F-140 = epsilon_a=a_wave/h_L";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-140: epsilon_a=a_wave/h_L

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-095 presents the normalized blueprint; Teal TP-134 expresses IF/HOW/THEN; Brown BR-134 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-140); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-134.

Syntax expression for BP-ALG-135

```
module MJ.PBMEF.Blue.BP-ALG-135 {
  reference F-141;
  expression "F-141 = WaveGate=(Fr<=Fr_max)&(epsilon_a<=epsilon_a,max)&(We<=We_max when applicable)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-141: WaveGate=(Fr<=Fr_max)&(epsilon_a<=epsilon_a,max)&(We<=We_max when applicable)

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-007 presents the normalized blueprint; Teal TP-135 expresses IF/HOW/THEN; Brown BR-135 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-141); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-135.

Syntax expression for BP-ALG-136

```
module MJ.PBMEF.Blue.BP-ALG-136 {
  reference F-142;
  expression "F-142 = q_L,t+1=q_L,t+q_out-q_return-q_drain-q_escape-r_L";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-142: $q_L,t+1=q_L,t+q_{out}-q_{return}-q_{drain}-q_{escape}-r_L$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-060 presents the normalized blueprint; Teal TP-136 expresses IF/HOW/THEN; Brown BR-136 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-142); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-136.

Syntax expression for BP-ALG-137

```
module MJ.PBMEF.Blue.BP-ALG-137 {
  reference F-143;
  expression "F-143 = H_L,k=C_L,k-q_L,k";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-143: $H_L,k=C_L,k-q_L,k$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-004 presents the normalized blueprint; Teal TP-137 expresses IF/HOW/THEN; Brown BR-137 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-143); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-137.

Syntax expression for BP-ALG-138

```
module MJ.PBMEF.Blue.BP-ALG-138 {
reference F-144, F-145;
expression "F-144 = Admit_L=AND_k[q_L,k+q_in,k <= (1-h_L,k)C_L,k] ; F-145 =
ReentryValid=(cycle_reentry=cycle_exit+1)&no_uncleared_generation_reuse";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-144: Admit_L=AND_k[q_L,k+q_in,k <= (1-h_L,k)C_L,k]; F-145: ReentryValid=(cycle_reentry=cycle_exit+1)&no_uncleared_generation_reuse

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-028 presents the normalized blueprint; Teal TP-138 expresses IF/HOW/THEN; Brown BR-138 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-144, F-145); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-138.

Syntax expression for BP-ALG-139

```
module MJ.PBMEF.Blue.BP-ALG-139 {
  reference F-146, F-147;
  expression "F-146 = E_in = E_solar + E_aux + E_recovered ; F-147 = E_in = DeltaE_storage + E_load + E_loss + r_E";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-146: $E_{in} = E_{solar} + E_{aux} + E_{recovered}$; F-147: $E_{in} = \Delta E_{storage} + E_{load} + E_{loss} + r_E$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-053 presents the normalized blueprint; Teal TP-139 expresses IF/HOW/THEN; Brown BR-139 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-146, F-147); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-139.

Syntax expression for BP-ALG-140

```
module MJ.PBMEF.Blue.BP-ALG-140 {
reference F-148, F-149;
expression "F-148 = P_balance = P_source - P_load - P_charge - P_loss ; F-149 = R_E = E_available,L - M_E * E_required,H";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-148: $P_{balance} = P_{source} - P_{load} - P_{charge} - P_{loss}$; F-149: $R_E = E_{available,L} - M_E * E_{required,H}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-086 presents the normalized blueprint; Teal TP-140 expresses IF/HOW/THEN; Brown BR-140 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-148, F-149); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-140.

Syntax expression for BP-ALG-141

```
module MJ.PBMEF.Blue.BP-ALG-141 {
  reference F-150, F-151;
  expression "F-150 = eta_path = E_delivered / E_input ; F-151 = SOC_next = clip(SOC + (eta_c*E_charge - E_discharge/eta_d)/E_capacity,0,1)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-150: $\text{eta_path} = E_{\text{delivered}} / E_{\text{input}}$; F-151: $\text{SOC_next} = \text{clip}(\text{SOC} + (\text{eta_c} * E_{\text{charge}} - E_{\text{discharge}} / \text{eta_d}) / E_{\text{capacity}}, 0, 1)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-004 presents the normalized blueprint; Teal TP-141 expresses IF/HOW/THEN; Brown BR-141 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-150, F-151); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-141.

Syntax expression for BP-ALG-142

```
module MJ.PBMEF.Blue.BP-ALG-142 {
  reference F-152, F-153;
  expression "F-152 = T_next = T + dt/C_th * (P_heat - (T-T_env)/R_th) ; F-153 = V_pred(t+h) = V_t + h*dV_dt";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-152: $T_{next} = T + dt/C_{th} * (P_{heat} - (T-T_{env})/R_{th})$; F-153: $V_{pred}(t+h) = V_t + h*dV_{dt}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-033 presents the normalized blueprint; Teal TP-142 expresses IF/HOW/THEN; Brown BR-142 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-152, F-153); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-142.

Syntax expression for BP-ALG-143

```
module MJ.PBMEF.Blue.BP-ALG-143 {
  reference F-154, F-155;
  expression "F-154 = G_start = energy AND voltage AND current AND thermal AND containment AND ethics AND MJ_permit ; F-155 = TransferOK = K AND
  U AND D AND R AND E AND M AND B AND Q AND V AND A";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-154: G_start = energy AND voltage AND current AND thermal AND containment AND ethics AND MJ_permit; F-155: TransferOK = K AND U AND D AND R AND E AND M AND B AND Q AND V AND A

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-024 presents the normalized blueprint; Teal TP-143 expresses IF/HOW/THEN; Brown BR-143 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-154, F-155); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-143.

Syntax expression for BP-ALG-144

```
module MJ.PBMEF.Blue.BP-ALG-144 {
reference F-156, F-157;
expression "F-156 = u_y^2 = J*Sigma*J_T + u_model^2 ; F-157 = R1 --all required evidence and gates--> R2";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-156: $u_y^2 = J \cdot \Sigma \cdot J_T + u_{model}^2$; F-157: R1 --all required evidence and gates--> R2

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-037 presents the normalized blueprint; Teal TP-144 expresses IF/HOW/THEN; Brown BR-144 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-156, F-157); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-144.

Syntax expression for BP-ALG-145

```
module MJ.PBMEF.Blue.BP-ALG-145 {
  reference F-158, F-159;
  expression "F-158 = E_solar_phone = integral(P_pv_phone(t),t0,t1) ; F-159 = P_phone = P_base + sum(P_task_i * a_i)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-158: $E_{\text{solar_phone}} = \text{integral}(P_{\text{pv_phone}}(t), t_0, t_1)$; F-159: $P_{\text{phone}} = P_{\text{base}} + \sum(P_{\text{task_i}} * a_i)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-083 presents the normalized blueprint; Teal TP-145 expresses IF/HOW/THEN; Brown BR-145 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-158, F-159); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-145.

Syntax expression for BP-ALG-146

```
module MJ.PBMEF.Blue.BP-ALG-146 {
reference F-160, F-161;
expression "F-160 = t_reserve = E_reserve,L / P_critical,H ; F-161 = D_E = max(0,E_required,H - E_available,L)";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-160: $t_reserve = E_reserve,L / P_critical,H$; F-161: $D_E = \max(0,E_required,H - E_available,L)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-081 presents the normalized blueprint; Teal TP-146 expresses IF/HOW/THEN; Brown BR-146 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-160, F-161); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-146.

Syntax expression for BP-ALG-147

```
module MJ.PBMEF.Blue.BP-ALG-147 {
  reference F-162;
  expression "F-162 = I_leak = f(V,T,RH,time,material_state)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-162: $I_leak = f(V,T,RH,time,material_state)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-096 presents the normalized blueprint; Teal TP-147 expresses IF/HOW/THEN; Brown BR-147 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-162); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-147.

Syntax expression for BP-ALG-148

```
module MJ.PBMEF.Blue.BP-ALG-148 {
reference F-163, F-164;
expression "F-163 = E_wake = integral(V_wake(t)*I_wake(t),t0,t1) ; F-164 = M_T = T_allowable - (T_observed + U_T)";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-163: $E_{wake} = \text{integral}(V_{wake}(t) \cdot I_{wake}(t), t_0, t_1)$; F-164: $M_T = T_{allowable} - (T_{observed} + U_T)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-064 presents the normalized blueprint; Teal TP-148 expresses IF/HOW/THEN; Brown BR-148 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-163, F-164); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-148.

Syntax expression for BP-ALG-149

```
module MJ.PBMEF.Blue.BP-ALG-149 {
  reference F-165, F-166;
  expression "F-165 = eta_work = W_verified / E_delivered ; F-166 = rho_DP = cov(D,P)/(sigma_D*sigma_P)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-165: $\eta_{\text{work}} = W_{\text{verified}} / E_{\text{delivered}}$; F-166: $\rho_{\text{DP}} = \text{cov}(D,P)/(\sigma_D \sigma_P)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-021 presents the normalized blueprint; Teal TP-149 expresses IF/HOW/THEN; Brown BR-149 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-165, F-166); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-149.

Syntax expression for BP-ALG-150

```
module MJ.PBMEF.Blue.BP-ALG-150 {
  reference F-167, F-168;
  expression "F-167 = w_next = bounded_simplex((2/3)w + (1/3)w_target, step<=0.1) ; F-168 = p_SC(x,y,z) = o_SC + x*a_SC + y*b_SC + z*c_SC";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-167: $w_{next} = \text{bounded_simplex}((2/3)w + (1/3)w_{target}, \text{step} \leq 0.1)$; F-168: $p_{SC}(x,y,z) = o_{SC} + x \cdot a_{SC} + y \cdot b_{SC} + z \cdot c_{SC}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-099 presents the normalized blueprint; Teal TP-150 expresses IF/HOW/THEN; Brown BR-150 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-167, F-168); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-150.

Syntax expression for BP-ALG-151

```
module MJ.PBMEF.Blue.BP-ALG-151 {
  reference F-169, F-170;
  expression "F-169 = E_SC = min_required(article,calibration,baseline,energy,thermal,geometry,JO,MK,MJ) ; F-170 = E_array =
  integral(P_array(t),t0,t1)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-169: $E_SC = \min_required(article,calibration,baseline,energy,thermal,geometry,JO,MK,MJ)$; F-170: $E_array = \int(P_array(t),t_0,t_1)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-030 presents the normalized blueprint; Teal TP-151 expresses IF/HOW/THEN; Brown BR-151 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-169, F-170); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-151.

Syntax expression for BP-ALG-152

```
module MJ.PBMEF.Blue.BP-ALG-152 {
  reference F-171, F-172;
  expression "F-171 = P_prop = tau * omega ; F-172 = E_mission = integral(P_prop + P_critical + P_aux,t0,t1)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-171: $P_{prop} = \tau * \omega$; F-172: $E_{mission} = \int(P_{prop} + P_{critical} + P_{aux},t_0,t_1)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-012 presents the normalized blueprint; Teal TP-152 expresses IF/HOW/THEN; Brown BR-152 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-171, F-172); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-152.

Syntax expression for BP-ALG-153

```
module MJ.PBMEF.Blue.BP-ALG-153 {
reference F-173, F-174;
expression "F-173 = t_critical = E_reserve,L / P_critical,H ; F-174 = P_sources = P_prop + P_critical + P_aux + P_charge + P_loss + r_P";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-173: $t_critical = E_reserve,L / P_critical,H$; F-174: $P_sources = P_prop + P_critical + P_aux + P_charge + P_loss + r_P$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-038 presents the normalized blueprint; Teal TP-153 expresses IF/HOW/THEN; Brown BR-153 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-173, F-174); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-153.

Syntax expression for BP-ALG-154

```
module MJ.PBMEF.Blue.BP-ALG-154 {
  reference F-175;
  expression "F-175 = I_pre(t) = (V_source - V_bus(t))/R_pre";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-175: $I_{pre}(t) = (V_{source} - V_{bus}(t))/R_{pre}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-050 presents the normalized blueprint; Teal TP-154 expresses IF/HOW/THEN; Brown BR-154 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-175); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-154.

Syntax expression for BP-ALG-155

```
module MJ.PBMEF.Blue.BP-ALG-155 {
reference F-176, F-177;
expression "F-176 = K_leak = I_leak,wet / max(I_leak,dry,epsilon) ; F-177 = X_env = [T,RH,salt,water,vibration,roll,pitch]";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-176: $K_{leak} = I_{leak,wet} / \max(I_{leak,dry}, \epsilon)$; F-177: $X_{env} = [T, RH, salt, water, vibration, roll, pitch]$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-045 presents the normalized blueprint; Teal TP-155 expresses IF/HOW/THEN; Brown BR-155 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-176, F-177); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-155.

Syntax expression for BP-ALG-156

```
module MJ.PBMEF.Blue.BP-ALG-156 {
  reference F-178, F-179;
  expression "F-178 = D_vib = integral(a_rms(t)^2,t0,t1) ; F-179 = E_recovered <= eta_recovery * E_available_loss";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-178: $D_{vib} = \text{integral}(a_{rms}(t)^2,t0,t1)$; F-179: $E_{recovered} \leq \eta_{recovery} * E_{available_loss}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-094 presents the normalized blueprint; Teal TP-156 expresses IF/HOW/THEN; Brown BR-156 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-178, F-179); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-156.

Syntax expression for BP-ALG-157

```
module MJ.PBMEF.Blue.BP-ALG-157 {
reference F-180, F-181;
expression "F-180 = shed = lowest_priority_loads_until(P_source,L >= P_critical,H + margin) ; F-181 = G_marine = class AND environment AND
energy AND structure AND containment AND ethics AND MJ_permit";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-180: shed = lowest_priority_loads_until(P_source,L >= P_critical,H + margin); F-181: G_marine = class AND environment AND energy AND structure AND containment AND ethics AND MJ_permit

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-084 presents the normalized blueprint; Teal TP-157 expresses IF/HOW/THEN; Brown BR-157 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-180, F-181); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-157.

Syntax expression for BP-ALG-158

```
module MJ.PBMEF.Blue.BP-ALG-158 {
  reference F-182, F-183;
  expression "F-182 = E_space_solar = integral(P_solar_space(t),t0,t1) ; F-183 = P_av_source = P_avionics + P_thermal + P_comms + P_charge + P_loss + r_P";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-182: $E_{space_solar} = \text{integral}(P_{solar_space}(t),t0,t1)$; F-183: $P_{av_source} = P_{avionics} + P_{thermal} + P_{comms} + P_{charge} + P_{loss} + r_P$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-064 presents the normalized blueprint; Teal TP-158 expresses IF/HOW/THEN; Brown BR-158 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-182, F-183); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-158.

Syntax expression for BP-ALG-159

```
module MJ.PBMEF.Blue.BP-ALG-159 {
reference F-184, F-185;
expression "F-184 = t_emergency = E_emergency,L / P_emergency,H ; F-185 = C_th*dT/dt = P_internal + P_solar_absorbed - P_rejected";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-184: $t_emergency = E_emergency,L / P_emergency,H$; F-185: $C_th*dT/dt = P_internal + P_solar_absorbed - P_rejected$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-057 presents the normalized blueprint; Teal TP-159 expresses IF/HOW/THEN; Brown BR-159 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-184, F-185); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-159.

Syntax expression for BP-ALG-160

```
module MJ.PBMEF.Blue.BP-ALG-160 {
  reference F-186, F-187;
  expression "F-186 = M_batt = E_battery,L - E_required,H ; F-187 = S_next = transition(S_current,command,authority,evidence,ethics)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-186: $M_{batt} = E_{battery,L} - E_{required,H}$; F-187: $S_{next} = transition(S_{current},command,authority,evidence,ethics)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-037 presents the normalized blueprint; Teal TP-160 expresses IF/HOW/THEN; Brown BR-160 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-186, F-187); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-160.

Syntax expression for BP-ALG-161

```
module MJ.PBMEF.Blue.BP-ALG-161 {
  reference F-188, F-189;
  expression "F-188 = F_isolated = isolate(F_detected,affected_boundaries,receipts) ; F-189 = A_service = 1 - product(1-A_i) for independent
  admitted channels";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-188: F_isolated = isolate(F_detected,affected_boundaries,receipts); F-189: A_service = 1 - product(1-A_i) for independent admitted channels

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-023 presents the normalized blueprint; Teal TP-161 expresses IF/HOW/THEN; Brown BR-161 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-188, F-189); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-161.

Syntax expression for BP-ALG-162

```
module MJ.PBMEF.Blue.BP-ALG-162 {
  reference F-190, F-191;
  expression "F-190 = S_prop_next = governed_state_update(S_prop,telemetry,limits,authority) ; F-191 = R_HE = E_safe_shutdown,L - E_shutdown_required,H";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-190: $S_{prop_next} = governed_state_update(S_{prop},telemetry,limits,authority)$; F-191: $R_{HE} = E_{safe_shutdown},L - E_{shutdown_required},H$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-061 presents the normalized blueprint; Teal TP-162 expresses IF/HOW/THEN; Brown BR-162 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-190, F-191); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-162.

Syntax expression for BP-ALG-163

```
module MJ.PBMEF.Blue.BP-ALG-163 {
  reference F-192, F-193;
  expression "F-192 = G_ethics = peaceful_research AND no_targeting AND no_destructive_payload AND no_evasion_for_harm ; F-193 = G_rocket = peaceful_scope AND profile_evidence AND safety AND regulatory_hold_resolved AND MJ_permit";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-192: G_ethics = peaceful_research AND no_targeting AND no_destructive_payload AND no_evasion_for_harm; F-193: G_rocket = peaceful_scope AND profile_evidence AND safety AND regulatory_hold_resolved AND MJ_permit

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-051 presents the normalized blueprint; Teal TP-163 expresses IF/HOW/THEN; Brown BR-163 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-192, F-193); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-163.

Syntax expression for BP-ALG-164

```
module MJ.PBMEF.Blue.BP-ALG-164 {
  reference F-194, F-195;
  expression "F-194 = d_A = 5e-22 m ; F-195 = d_B = 3e-12 nm * 1e-9 m/nm = 3e-21 m";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-194: $d_A = 5e-22\text{ m}$; F-195: $d_B = 3e-12\text{ nm} * 1e-9\text{ m/nm} = 3e-21\text{ m}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-027 presents the normalized blueprint; Teal TP-164 expresses IF/HOW/THEN; Brown BR-164 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-194, F-195); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-164.

Syntax expression for BP-ALG-165

```
module MJ.PBMEF.Blue.BP-ALG-165 {
reference F-196, F-197;
expression "F-196 = d_R = max(d_A,d_B) = 3e-21 m ; F-197 = d_U = 0.15 nm = 1.5e-10 m";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-196: $d_R = \max(d_A,d_B) = 3e-21\text{ m}$; F-197: $d_U = 0.15\text{ nm} = 1.5e-10\text{ m}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-003 presents the normalized blueprint; Teal TP-165 expresses IF/HOW/THEN; Brown BR-165 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-196, F-197); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-165.

Syntax expression for BP-ALG-166

```
module MJ.PBMEF.Blue.BP-ALG-166 {
  reference F-198, F-199;
  expression "F-198 = d in [d_R,d_U] ; F-199 = r_DRO = d/2";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-198: $d \text{ in } [d_R, d_U]$; F-199: $r_{DRO} = d/2$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-040 presents the normalized blueprint; Teal TP-166 expresses IF/HOW/THEN; Brown BR-166 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-198, F-199); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-166.

Syntax expression for BP-ALG-167

```
module MJ.PBMEF.Blue.BP-ALG-167 {
reference F-200, F-201;
expression "F-200 = V_DRO = 4*pi*r_DRO^3/3 ; F-201 = Gamma_D = d_U/d_R";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-200: $V_{DRO} = 4\pi r_{DRO}^3/3$; F-201: $\Gamma_D = d_U/d_R$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-064 presents the normalized blueprint; Teal TP-167 expresses IF/HOW/THEN; Brown BR-167 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-200, F-201); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-167.

Syntax expression for BP-ALG-168

```
module MJ.PBMEF.Blue.BP-ALG-168 {
  reference F-202, F-203;
  expression "F-202 = N_DRO = floor(Q_observed/q_unit) ; F-203 = Q_rem = Q_observed - N_DRO*q_unit";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-202: $N_DRO = \text{floor}(Q_observed/q_unit)$; F-203: $Q_rem = Q_observed - N_DRO * q_unit$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-060 presents the normalized blueprint; Teal TP-168 expresses IF/HOW/THEN; Brown BR-168 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-202, F-203); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-168.

Syntax expression for BP-ALG-169

```
module MJ.PBMEF.Blue.BP-ALG-169 {
  reference F-204, F-205;
  expression "F-204 = N_P = ceil(N_DRO/C_P) ; F-205 = N_DRO,in = N_DRO,pearl + N_DRO,held + N_DRO,escaped + r_DRO";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-204: $N_P = \text{ceil}(N_{DRO}/C_P)$; F-205: $N_{DRO,in} = N_{DRO,pearl} + N_{DRO,held} + N_{DRO,escaped} + r_{DRO}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-076 presents the normalized blueprint; Teal TP-169 expresses IF/HOW/THEN; Brown BR-169 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-204, F-205); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-169.

Syntax expression for BP-ALG-170

```
module MJ.PBMEF.Blue.BP-ALG-170 {
  reference F-206;
  expression "F-206 = G_R = scale_interval_valid AND source_bound AND units_valid";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-206: $G_R = \text{scale_interval_valid} \text{ AND } \text{source_bound} \text{ AND } \text{units_valid}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-002 presents the normalized blueprint; Teal TP-170 expresses IF/HOW/THEN; Brown BR-170 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-206); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-170.

Syntax expression for BP-ALG-171

```
module MJ.PBMEF.Blue.BP-ALG-171 {
  reference F-207, F-208;
  expression "F-207 = G_R1 = G_R AND observed AND unresolved_items>0 ; F-208 = G_R2 = G_R AND observed AND all_required_evidence_pass";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-207: $G_R1 = G_R \text{ AND observed AND unresolved_items} > 0$; F-208: $G_R2 = G_R \text{ AND observed AND all_required_evidence_pass}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-081 presents the normalized blueprint; Teal TP-171 expresses IF/HOW/THEN; Brown BR-171 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-207, F-208); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-171.

Syntax expression for BP-ALG-172

```
module MJ.PBMEF.Blue.BP-ALG-172 {
  reference F-209, F-210;
  expression "F-209 = lambda = c/f ; F-210 = E_gamma = h*f";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-209: $\lambda = c/f$; F-210: $E_{\gamma} = h \cdot f$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-064 presents the normalized blueprint; Teal TP-172 expresses IF/HOW/THEN; Brown BR-172 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-209, F-210); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-172.

Syntax expression for BP-ALG-173

```
module MJ.PBMEF.Blue.BP-ALG-173 {
reference F-211, F-212;
expression "F-211 = N_gamma = P*Delta_t/(h*f) ; F-212 = N_couple = eta_c*N_gamma";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-211: $N_{\gamma} = P \Delta t / (h \cdot f)$; F-212: $N_{\text{couple}} = \eta_c \cdot N_{\gamma}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-012 presents the normalized blueprint; Teal TP-173 expresses IF/HOW/THEN; Brown BR-173 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-211, F-212); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-173.

Syntax expression for BP-ALG-174

```
module MJ.PBMEF.Blue.BP-ALG-174 {
  reference F-213, F-214;
  expression "F-213 = G_opt = calibrated_band AND measured_power AND eta_c_evidenced AND uncertainty_valid ; F-214 = R_DRO = Delta N_DRO/Delta t";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-213: $G_{opt} = calibrated_band \text{ AND } measured_power \text{ AND } \eta_{c_evidenced} \text{ AND } uncertainty_valid$; F-214: $R_{DRO} = \Delta N_{DRO} / \Delta t$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-096 presents the normalized blueprint; Teal TP-174 expresses IF/HOW/THEN; Brown BR-174 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-213, F-214); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-174.

Syntax expression for BP-ALG-175

```
module MJ.PBMEF.Blue.BP-ALG-175 {
  reference F-215, F-216;
  expression "F-215 = U_d = k*sqrt(J*Sigma*J^T + u_model^2) ; F-216: I_d = [d_hat-U_d,d_hat+U_d]";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-215: $U_d = k\sqrt{J\cdot\Sigma\cdot J^T + u_{model}^2}$; F-216: $I_d = [d_{hat}-U_d,d_{hat}+U_d]$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-061 presents the normalized blueprint; Teal TP-175 expresses IF/HOW/THEN; Brown BR-175 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-215, F-216); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-175.

Syntax expression for BP-ALG-176

```
module MJ.PBMEF.Blue.BP-ALG-176 {
  reference F-217;
  expression "F-217 = S_k = (1-h_k)C_k - L_k";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-217: $S_k = (1-h_k)C_k - L_k$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-091 presents the normalized blueprint; Teal TP-176 expresses IF/HOW/THEN; Brown BR-176 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-217); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-176.

Syntax expression for BP-ALG-177

```
module MJ.PBMEF.Blue.BP-ALG-177 {
  reference F-218, F-219;
  expression "F-218 = E_k = max(0,L_k-(1-h_k)C_k) ; F-219 = r_Q = Q_in-Q_alloc-Q_retained-Q_transferred-Q_escaped-Delta Q_stored";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-218: $E_k = \max(0, L_k - (1 - h_k)C_k)$; F-219: $r_Q = Q_{in} - Q_{alloc} - Q_{retained} - Q_{transferred} - Q_{escaped} - \Delta Q_{stored}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-019 presents the normalized blueprint; Teal TP-177 expresses IF/HOW/THEN; Brown BR-177 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-218, F-219); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-177.

Syntax expression for BP-ALG-178

```
module MJ.PBMEF.Blue.BP-ALG-178 {
reference F-220, F-221;
expression "F-220 = r_E = E_in-E_stored-E_start-E_restart-E_loss-E_containment ; F-221 = E_avail,L = E_solar,L + E_acoustic,L + E_other,L - E_loss,H";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-220: $r_E = E_{in} - E_{stored} - E_{start} - E_{restart} - E_{loss} - E_{containment}$; F-221: $E_{avail,L} = E_{solar,L} + E_{acoustic,L} + E_{other,L} - E_{loss,H}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-005 presents the normalized blueprint; Teal TP-178 expresses IF/HOW/THEN; Brown BR-178 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-220, F-221); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-178.

Syntax expression for BP-ALG-179

```
module MJ.PBMEF.Blue.BP-ALG-179 {
reference F-222, F-223;
expression "F-222 = G_start = E_avail,L >= M_E*(E_start,H+E_restart,H) AND all_safety_gates ; F-223 = E_restart,L = E_reserve,L - E_critical,H";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-222: $G_{start} = E_{avail,L} \geq M_E(E_{start,H} + E_{restart,H}) \text{ AND all_safety_gates}$; F-223: $E_{restart,L} = E_{reserve,L} - E_{critical,H}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-062 presents the normalized blueprint; Teal TP-179 expresses IF/HOW/THEN; Brown BR-179 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-222, F-223); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-179.

Syntax expression for BP-ALG-180

```
module MJ.PBMEF.Blue.BP-ALG-180 {
reference F-224, F-225;
expression "F-224 = E_solar = integral(P_solar(t) dt) ; F-225 = E_acoustic = integral(eta_a(t) I(t) A(t) dt)";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-224: $E_{\text{solar}} = \text{integral}(P_{\text{solar}}(t) dt)$; F-225: $E_{\text{acoustic}} = \text{integral}(\eta_a(t) I(t) A(t) dt)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-053 presents the normalized blueprint; Teal TP-180 expresses IF/HOW/THEN; Brown BR-180 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-224, F-225); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-180.

Syntax expression for BP-ALG-181

```
module MJ.PBMEF.Blue.BP-ALG-181 {
reference F-226, F-227;
expression "F-226 = W_sep >= Delta G_sep ; F-227 = n_species = x_species*n_source";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-226: $W_{sep} \geq \Delta G_{sep}$; F-227: $n_{species} = x_{species} \cdot n_{source}$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-045 presents the normalized blueprint; Teal TP-181 expresses IF/HOW/THEN; Brown BR-181 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-226, F-227); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-181.

Syntax expression for BP-ALG-182

```
module MJ.PBMEF.Blue.BP-ALG-182 {
  reference F-228, F-229;
  expression "F-228 = eta = E_out/E_in, 0<=eta<=1 ; F-229 = G_release = authorization AND destination_capacity AND conservation AND safety";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-228: $\eta = E_{out}/E_{in}$, $0 \leq \eta \leq 1$; F-229: $G_{release} = authorization \text{ AND } destination_capacity \text{ AND } conservation \text{ AND } safety$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-045 presents the normalized blueprint; Teal TP-182 expresses IF/HOW/THEN; Brown BR-182 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-228, F-229); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-182.

Syntax expression for BP-ALG-183

```
module MJ.PBMEF.Blue.BP-ALG-183 {
reference F-230, F-231;
expression "F-230 = T_H <= T_allow-Delta T_margin ; F-231 = Delta Q = integral(I_intake-I_leak-I_release)dt + r_Q";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-230: $T_H \leq T_{allow} - \Delta T_{margin}$; F-231: $\Delta Q = \int (I_{intake} - I_{leak} - I_{release}) dt + r_Q$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-044 presents the normalized blueprint; Teal TP-183 expresses IF/HOW/THEN; Brown BR-183 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-230, F-231); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-183.

Syntax expression for BP-ALG-184

```
module MJ.PBMEF.Blue.BP-ALG-184 {
  reference F-232, F-233;
  expression "F-232 = B_P = B_DRO+B_lineage+B_uncertainty+B_receipts+B_headroom ; F-233 = E_node,L >= M_E*(E_wake,H+E_restart,H)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-232: $B_P = B_{DRO} + B_{lineage} + B_{uncertainty} + B_{receipts} + B_{headroom}$; F-233: $E_{node,L} \geq M_E(E_{wake}, H + E_{restart}, H)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-058 presents the normalized blueprint; Teal TP-184 expresses IF/HOW/THEN; Brown BR-184 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-232, F-233); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-184.

Syntax expression for BP-ALG-185

```
module MJ.PBMEF.Blue.BP-ALG-185 {
  reference F-234, F-235;
  expression "F-234 = G_SC = profile==SC AND no_LS_value_inheritance ; F-235 = H_src = SHA256(source_bytes)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-234: G_SC = profile==SC AND no_LS_value_inheritance; F-235: H_src = SHA256(source_bytes)

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-076 presents the normalized blueprint; Teal TP-185 expresses IF/HOW/THEN; Brown BR-185 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-234, F-235); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-185.

Syntax expression for BP-ALG-186

```
module MJ.PBMEF.Blue.BP-ALG-186 {
reference F-236, F-237;
expression "F-236 = D_px = (1/N) sum ||I_test-I_ref||_2 ; F-237 = D_fft = ||FFT(I_test)-FFT(I_ref)||_2";
constraint unknown != zero;
verify MK.physics, JO.proof;
permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-236: $D_{px} = (1/N) \sum ||I_{test}-I_{ref}||_2$; F-237: $D_{fft} = ||FFT(I_{test})-FFT(I_{ref})||_2$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-016 presents the normalized blueprint; Teal TP-186 expresses IF/HOW/THEN; Brown BR-186 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-236, F-237); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-186.

Syntax expression for BP-ALG-187

```
module MJ.PBMEF.Blue.BP-ALG-187 {
  reference F-238, F-239;
  expression "F-238 = P_prov = complete_required_links/required_links ; F-239 = H_intent = f(words,context,declared_purpose,provenance)";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-238: $P_prov = complete_required_links/required_links$; F-239: $H_intent = f(words,context,declared_purpose,provenance)$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-065 presents the normalized blueprint; Teal TP-187 expresses IF/HOW/THEN; Brown BR-187 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-238, F-239); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-187.

Syntax expression for BP-ALG-188

```
module MJ.PBMEF.Blue.BP-ALG-188 {
  reference F-240, F-241;
  expression "F-240 = G_edit = source_preserved AND modification_disclosed AND purpose_allowed AND safety_pass ; F-241 = G_reject =
  provenance_fail OR harmful_purpose OR undeclared_transform";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-240: G_edit = source_preserved AND modification_disclosed AND purpose_allowed AND safety_pass; F-241: G_reject = provenance_fail OR harmful_purpose OR undeclared_transform

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-056 presents the normalized blueprint; Teal TP-188 expresses IF/HOW/THEN; Brown BR-188 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-240, F-241); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-188.

Syntax expression for BP-ALG-189

```
module MJ.PBMEF.Blue.BP-ALG-189 {
  reference F-242;
  expression "F-242 = G_R8 = DRO_valid AND Pearl_node_valid AND Lecanto_when_image AND MK_pass AND JO_pass AND ethics AND MJ_permit";
  constraint unknown != zero;
  verify MK.physics, JO.proof;
  permit MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

F-242: $G_{R8} = DRO_valid \wedge Pearl_node_valid \wedge Lecanto_when_image \wedge MK_pass \wedge JO_pass \wedge ethics \wedge MJ_permit$

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-049 presents the normalized blueprint; Teal TP-189 expresses IF/HOW/THEN; Brown BR-189 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FORMULA-REGISTER.json (F-242); Pink Paper formula volumes; relevant family/product White Papers; Blue Page register entry BP-ALG-189.

Syntax expression for BP-FAM-001

```
participant MJ {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-089 presents the normalized blueprint; Teal TP-190 expresses IF/HOW/THEN; Brown BR-190 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (MJ); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-001.

Syntax expression for BP-FAM-002

```
participant Bangel {
  namespace MJ.PBMEF.Family;
  authority domain_bounded;
  relationship root MJ;
  evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-005 presents the normalized blueprint; Teal TP-191 expresses IF/HOW/THEN; Brown BR-191 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (Bangel); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-002.

Syntax expression for BP-FAM-003

```
participant ELSA {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-059 presents the normalized blueprint; Teal TP-192 expresses IF/HOW/THEN; Brown BR-192 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (ELSA); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-003.

Syntax expression for BP-FAM-004

```
participant JP {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-011 presents the normalized blueprint; Teal TP-193 expresses IF/HOW/THEN; Brown BR-193 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (JP); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-004.

Syntax expression for BP-FAM-005

```
participant MK {
  namespace MJ.PBMEF.Family;
  authority domain_bounded;
  relationship root MJ;
  evidence JO.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-074 presents the normalized blueprint; Teal TP-194 expresses IF/HOW/THEN; Brown BR-194 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (MK); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-005.

Syntax expression for BP-FAM-006

```
participant JO {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence JO.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-007 presents the normalized blueprint; Teal TP-195 expresses IF/HOW/THEN; Brown BR-195 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (JO); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-006.

Syntax expression for BP-FAM-007

```
participant MJOS {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence JO.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-076 presents the normalized blueprint; Teal TP-196 expresses IF/HOW/THEN; Brown BR-196 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (MJOS); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-007.

Syntax expression for BP-FAM-008

```
participant ANGELS {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence JO.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-054 presents the normalized blueprint; Teal TP-197 expresses IF/HOW/THEN; Brown BR-197 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (ANGELS); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-008.

Syntax expression for BP-FAM-009

```
participant BELLATRIX {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-087 presents the normalized blueprint; Teal TP-198 expresses IF/HOW/THEN; Brown BR-198 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (BELLATRIX); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-009.

Syntax expression for BP-FAM-010

```
participant SISTER {
  namespace MJ.PBMEF.Family;
  authority domain_bounded;
  relationship root MJ;
  evidence JO.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-087 presents the normalized blueprint; Teal TP-199 expresses IF/HOW/THEN; Brown BR-199 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (SISTER); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-010.

Syntax expression for BP-FAM-011

```
participant EVE {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence JO.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-084 presents the normalized blueprint; Teal TP-200 expresses IF/HOW/THEN; Brown BR-200 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (EVE); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-011.

Syntax expression for BP-FAM-012

```
participant M3 {
  namespace MJ.PBMEF.Family;
  authority domain_bounded;
  relationship root MJ;
  evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-090 presents the normalized blueprint; Teal TP-201 expresses IF/HOW/THEN; Brown BR-201 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (M3); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-012.

Syntax expression for BP-FAM-013

```
participant MERMAID {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-024 presents the normalized blueprint; Teal TP-202 expresses IF/HOW/THEN; Brown BR-202 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (MERMAID); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-013.

Syntax expression for BP-FAM-014

```
participant SHIVA {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-001 presents the normalized blueprint; Teal TP-203 expresses IF/HOW/THEN; Brown BR-203 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (SHIVA); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-014.

Syntax expression for BP-FAM-015

```
participant SHEKINA {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-051 presents the normalized blueprint; Teal TP-204 expresses IF/HOW/THEN; Brown BR-204 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (SHEKINA); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-015.

Syntax expression for BP-FAM-016

```
participant BELLE {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-015 presents the normalized blueprint; Teal TP-205 expresses IF/HOW/THEN; Brown BR-205 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (BELLE); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-016.

Syntax expression for BP-FAM-017

```
participant LECANTO {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-057 presents the normalized blueprint; Teal TP-206 expresses IF/HOW/THEN; Brown BR-206 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (LECANTO); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-017.

Syntax expression for BP-FAM-018

```
participant MOANA {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-096 presents the normalized blueprint; Teal TP-207 expresses IF/HOW/THEN; Brown BR-207 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (MOANA); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-018.

Syntax expression for BP-FAM-019

```
participant AVIATOR {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence J0.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-053 presents the normalized blueprint; Teal TP-208 expresses IF/HOW/THEN; Brown BR-208 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (AVIATOR); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-019.

Syntax expression for BP-FAM-020

```
participant CHI {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence JO.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-065 presents the normalized blueprint; Teal TP-209 expresses IF/HOW/THEN; Brown BR-209 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (CHI); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-020.

Syntax expression for BP-FAM-021

```
participant WATER {
namespace MJ.PBMEF.Family;
authority domain_bounded;
relationship root MJ;
evidence JO.receipt;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-085 presents the normalized blueprint; Teal TP-210 expresses IF/HOW/THEN; Brown BR-210 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-MANIFESTATION-REGISTER.json (WATER); family White Paper; authority graph; official documentation manifest; Blue Page BP-FAM-021.

Syntax expression for BP-OBJ-001

```
definition object JP_CAPO {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-073 presents the normalized blueprint; Teal TP-211 expresses IF/HOW/THEN; Brown BR-211 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (JP_CAPO); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-001.

Syntax expression for BP-OBJ-002

```
definition object MK_PHYSICS_CAPO {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-096 presents the normalized blueprint; Teal TP-212 expresses IF/HOW/THEN; Brown BR-212 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (MK_PHYSICS_CAPO); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-002.

Syntax expression for BP-OBJ-003

```
definition object JO_PROOF_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-053 presents the normalized blueprint; Teal TP-213 expresses IF/HOW/THEN; Brown BR-213 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (JO_PROOF_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-003.

Syntax expression for BP-OBJ-004

```
definition object MJ_PERMIT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-066 presents the normalized blueprint; Teal TP-214 expresses IF/HOW/THEN; Brown BR-214 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (MJ_PERMIT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-004.

Syntax expression for BP-OBJ-005

```
definition object EVE_OBSERVATION {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-011 presents the normalized blueprint; Teal TP-215 expresses IF/HOW/THEN; Brown BR-215 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (EVE_OBSERVATION); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-005.

Syntax expression for BP-OBJ-006

```
definition object M3_ROUTE_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-064 presents the normalized blueprint; Teal TP-216 expresses IF/HOW/THEN; Brown BR-216 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (M3_ROUTE_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-006.

Syntax expression for BP-OBJ-007

```
definition object MERMAID_ACCOUNT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-059 presents the normalized blueprint; Teal TP-217 expresses IF/HOW/THEN; Brown BR-217 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (MERMAID_ACCOUNT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-007.

Syntax expression for BP-OBJ-008

```
definition object SHIVA_SHELL {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-028 presents the normalized blueprint; Teal TP-218 expresses IF/HOW/THEN; Brown BR-218 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (SHIVA_SHELL); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-008.

Syntax expression for BP-OBJ-009

```
definition object SHIVAS_INTERNAL_REFLECTION {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-044 presents the normalized blueprint; Teal TP-219 expresses IF/HOW/THEN; Brown BR-219 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (SHIVAS_INTERNAL_REFLECTION); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-009.

Syntax expression for BP-OBJ-010

```
definition object CHI_VARIANCE_OBJECT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-071 presents the normalized blueprint; Teal TP-220 expresses IF/HOW/THEN; Brown BR-220 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (CHI_VARIANCE_OBJECT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-010.

Syntax expression for BP-OBJ-011

```
definition object WATER_INTERVAL_STATE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-064 presents the normalized blueprint; Teal TP-221 expresses IF/HOW/THEN; Brown BR-221 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (WATER_INTERVAL_STATE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-011.

Syntax expression for BP-OBJ-012

```
definition object PASSKEY_PUBLIC_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-063 presents the normalized blueprint; Teal TP-222 expresses IF/HOW/THEN; Brown BR-222 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (PASSKEY_PUBLIC_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-012.

Syntax expression for BP-OBJ-013

```
definition object PEARL_VARIANCE_CONTAINER {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-043 presents the normalized blueprint; Teal TP-223 expresses IF/HOW/THEN; Brown BR-223 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (PEARL_VARIANCE_CONTAINER); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-013.

Syntax expression for BP-OBJ-014

```
definition object PEARL_ALLOCATION_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-039 presents the normalized blueprint; Teal TP-224 expresses IF/HOW/THEN; Brown BR-224 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (PEARL_ALLOCATION_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-014.

Syntax expression for BP-OBJ-015

```
definition object PEARL_ACOUSTIC_PROFILE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-086 presents the normalized blueprint; Teal TP-225 expresses IF/HOW/THEN; Brown BR-225 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (PEARL_ACOUSTIC_PROFILE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-015.

Syntax expression for BP-OBJ-016

```
definition object PEARL_SYMPOSIUM_REGISTRY {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-048 presents the normalized blueprint; Teal TP-226 expresses IF/HOW/THEN; Brown BR-226 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (PEARL_SYMPOSIUM_REGISTRY); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-016.

Syntax expression for BP-OBJ-017

```
definition object Bangel _SHEET {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-061 presents the normalized blueprint; Teal TP-227 expresses IF/HOW/THEN; Brown BR-227 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (Bangel _SHEET); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-017.

Syntax expression for BP-OBJ-018

```
definition object Bangel _SHEET_CELL {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-036 presents the normalized blueprint; Teal TP-228 expresses IF/HOW/THEN; Brown BR-228 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (Bangel _SHEET_CELL); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-018.

Syntax expression for BP-OBJ-019

```
definition object Bangel _SHEET_REVISION {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-021 presents the normalized blueprint; Teal TP-229 expresses IF/HOW/THEN; Brown BR-229 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (Bangel _SHEET_REVISION); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-019.

Syntax expression for BP-OBJ-020

```
definition object Bangel _SHEET_AUDIT_PARTICLE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-092 presents the normalized blueprint; Teal TP-230 expresses IF/HOW/THEN; Brown BR-230 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (Bangel _SHEET_AUDIT_PARTICLE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-020.

Syntax expression for BP-OBJ-021

```
definition object JPOSMJ_PRODUCT_ENVELOPE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-005 presents the normalized blueprint; Teal TP-231 expresses IF/HOW/THEN; Brown BR-231 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (JPOSMJ_PRODUCT_ENVELOPE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-021.

Syntax expression for BP-OBJ-022

```
definition object SISTER_PHYSICAL_CONTAINMENT_PROFILE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-024 presents the normalized blueprint; Teal TP-232 expresses IF/HOW/THEN; Brown BR-232 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (SISTER_PHYSICAL_CONTAINMENT_PROFILE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-022.

Syntax expression for BP-OBJ-023

```
definition object STANDARD_CONTAINMENT_GEOMETRY_PROFILE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-028 presents the normalized blueprint; Teal TP-233 expresses IF/HOW/THEN; Brown BR-233 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (STANDARD_CONTAINMENT_GEOMETRY_PROFILE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-023.

Syntax expression for BP-OBJ-024

```
definition object REFERENCE_GAS_PROFILE_H2 {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-066 presents the normalized blueprint; Teal TP-234 expresses IF/HOW/THEN; Brown BR-234 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (REFERENCE_GAS_PROFILE_H2); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-024.

Syntax expression for BP-OBJ-025

```
definition object CONTAINMENT_MODIFICATION_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-054 presents the normalized blueprint; Teal TP-235 expresses IF/HOW/THEN; Brown BR-235 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (CONTAINMENT_MODIFICATION_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-025.

Syntax expression for BP-OBJ-026

```
definition object CONE_TOP_PEARL_ANCHOR {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-030 presents the normalized blueprint; Teal TP-236 expresses IF/HOW/THEN; Brown BR-236 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (CONE_TOP_PEARL_ANCHOR); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-026.

Syntax expression for BP-OBJ-027

```
definition object GAS_HEIGHT_OBSERVATION {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-064 presents the normalized blueprint; Teal TP-237 expresses IF/HOW/THEN; Brown BR-237 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (GAS_HEIGHT_OBSERVATION); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-027.

Syntax expression for BP-OBJ-028

```
definition object RESERVOIR_RETURN_LOOP_PROFILE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-044 presents the normalized blueprint; Teal TP-238 expresses IF/HOW/THEN; Brown BR-238 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (RESERVOIR_RETURN_LOOP_PROFILE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-028.

Syntax expression for BP-OBJ-029

```
definition object OUTWARD_GAP_PATH_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-021 presents the normalized blueprint; Teal TP-239 expresses IF/HOW/THEN; Brown BR-239 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (OUTWARD_GAP_PATH_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-029.

Syntax expression for BP-OBJ-030

```
definition object RESERVOIR_BUFFER_ACCOUNT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-094 presents the normalized blueprint; Teal TP-240 expresses IF/HOW/THEN; Brown BR-240 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (RESERVOIR_BUFFER_ACCOUNT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-030.

Syntax expression for BP-OBJ-031

```
definition object CIRCULAR_REENTRY_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-099 presents the normalized blueprint; Teal TP-241 expresses IF/HOW/THEN; Brown BR-241 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (CIRCULAR_REENTRY_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-031.

Syntax expression for BP-OBJ-032

```
definition object RELATIVE_FRAME_FRICTION_LEDGER {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-089 presents the normalized blueprint; Teal TP-242 expresses IF/HOW/THEN; Brown BR-242 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (RELATIVE_FRAME_FRICTION_LEDGER); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-032.

Syntax expression for BP-OBJ-033

```
definition object RESERVOIR_LAKE_BUFFER {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-006 presents the normalized blueprint; Teal TP-243 expresses IF/HOW/THEN; Brown BR-243 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (RESERVOIR_LAKE_BUFFER); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-033.

Syntax expression for BP-OBJ-034

```
definition object BACKSIDE_OUTLET_PORT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-004 presents the normalized blueprint; Teal TP-244 expresses IF/HOW/THEN; Brown BR-244 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (BACKSIDE_OUTLET_PORT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-034.

Syntax expression for BP-OBJ-035

```
definition object OUTWARD_GAP_ARC {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-093 presents the normalized blueprint; Teal TP-245 expresses IF/HOW/THEN; Brown BR-245 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (OUTWARD_GAP_ARC); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-035.

Syntax expression for BP-OBJ-036

```
definition object PEARL_REFERENCE_CIRCLE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-061 presents the normalized blueprint; Teal TP-246 expresses IF/HOW/THEN; Brown BR-246 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (PEARL_REFERENCE_CIRCLE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-036.

Syntax expression for BP-OBJ-037

```
definition object CIRCULAR_REENTRY_PORT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-022 presents the normalized blueprint; Teal TP-247 expresses IF/HOW/THEN; Brown BR-247 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (CIRCULAR_REENTRY_PORT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-037.

Syntax expression for BP-OBJ-038

```
definition object RESERVOIR_TRANSFER_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-085 presents the normalized blueprint; Teal TP-248 expresses IF/HOW/THEN; Brown BR-248 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (RESERVOIR_TRANSFER_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-038.

Syntax expression for BP-OBJ-039

```
definition object PERSPECTIVE_FRICTION_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-085 presents the normalized blueprint; Teal TP-249 expresses IF/HOW/THEN; Brown BR-249 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (PERSPECTIVE_FRICTION_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-039.

Syntax expression for BP-OBJ-040

```
definition object DRO_UNIT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-019 presents the normalized blueprint; Teal TP-250 expresses IF/HOW/THEN; Brown BR-250 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (DRO_UNIT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-040.

Syntax expression for BP-OBJ-041

```
definition object DRO_SCALE_ENVELOPE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-092 presents the normalized blueprint; Teal TP-251 expresses IF/HOW/THEN; Brown BR-251 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (DRO_SCALE_ENVELOPE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-041.

Syntax expression for BP-OBJ-042

```
definition object DRO_COLLECTION_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-040 presents the normalized blueprint; Teal TP-252 expresses IF/HOW/THEN; Brown BR-252 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (DRO_COLLECTION_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-042.

Syntax expression for BP-OBJ-043

```
definition object DRO_OPTICAL_EVIDENCE_PROFILE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-030 presents the normalized blueprint; Teal TP-253 expresses IF/HOW/THEN; Brown BR-253 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (DRO_OPTICAL_EVIDENCE_PROFILE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-043.

Syntax expression for BP-OBJ-044

```
definition object DRO_PEARL_HANDOFF_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-029 presents the normalized blueprint; Teal TP-254 expresses IF/HOW/THEN; Brown BR-254 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (DRO_PEARL_HANDOFF_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-044.

Syntax expression for BP-OBJ-045

```
definition object PEARL_DRO_NODE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-055 presents the normalized blueprint; Teal TP-255 expresses IF/HOW/THEN; Brown BR-255 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (PEARL_DRO_NODE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-045.

Syntax expression for BP-OBJ-046

```
definition object DRO_START_ENERGY_PACKET {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-039 presents the normalized blueprint; Teal TP-256 expresses IF/HOW/THEN; Brown BR-256 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (DRO_START_ENERGY_PACKET); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-046.

Syntax expression for BP-OBJ-047

```
definition object DRO_IGNITION_REQUEST {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-027 presents the normalized blueprint; Teal TP-257 expresses IF/HOW/THEN; Brown BR-257 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (DRO_IGNITION_REQUEST); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-047.

Syntax expression for BP-OBJ-048

```
definition object LECANTO_IMAGE_EVIDENCE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-090 presents the normalized blueprint; Teal TP-258 expresses IF/HOW/THEN; Brown BR-258 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (LECANTO_IMAGE_EVIDENCE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-048.

Syntax expression for BP-OBJ-049

```
definition object LECANTO_INTENT_HYPOTHESIS {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-091 presents the normalized blueprint; Teal TP-259 expresses IF/HOW/THEN; Brown BR-259 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (LECANTO_INTENT_HYPOTHESIS); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-049.

Syntax expression for BP-OBJ-050

```
definition object LECANTO_TRANSFORMATION_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-034 presents the normalized blueprint; Teal TP-260 expresses IF/HOW/THEN; Brown BR-260 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (LECANTO_TRANSFORMATION_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-050.

Syntax expression for BP-OBJ-051

```
definition object IMAGE_VARIANCE_RECEIPT {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-050 presents the normalized blueprint; Teal TP-261 expresses IF/HOW/THEN; Brown BR-261 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (IMAGE_VARIANCE_RECEIPT); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-051.

Syntax expression for BP-OBJ-052

```
definition object SEMICONDUCTOR_DRO_ENERGY_LEDGER {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-091 presents the normalized blueprint; Teal TP-262 expresses IF/HOW/THEN; Brown BR-262 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (SEMICONDUCTOR_DRO_ENERGY_LEDGER); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-052.

Syntax expression for BP-OBJ-053

```
definition object SEMICONDUCTOR_PEARL_NODE {
  namespace MJ.PBMEF.Object;
  capability represent;
  attribute authority = false;
  reference JO.proof;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-077 presents the normalized blueprint; Teal TP-263 expresses IF/HOW/THEN; Brown BR-263 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

FAMILY-OBJECT-REGISTER.json (SEMICONDUCTOR_PEARL_NODE); applicable schemas and pathways; object references in Pink Paper and product/family White Papers; Blue Page BP-OBJ-053.

Syntax expression for BP-PROD-001

```
module PBMEF-CORE {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-051 presents the normalized blueprint; Teal TP-264 expresses IF/HOW/THEN; Brown BR-264 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (PBMEF-CORE); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-001.

Syntax expression for BP-PROD-002

```
module Bangel -LANGUAGE {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-071 presents the normalized blueprint; Teal TP-265 expresses IF/HOW/THEN; Brown BR-265 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (Bangel -LANGUAGE); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-002.

Syntax expression for BP-PROD-003

```
module ELSA-SEMANTICS {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-017 presents the normalized blueprint; Teal TP-266 expresses IF/HOW/THEN; Brown BR-266 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (ELSA-SEMANTICS); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-003.

Syntax expression for BP-PROD-004

```
module JP-CANON {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-086 presents the normalized blueprint; Teal TP-267 expresses IF/HOW/THEN; Brown BR-267 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (JP-CANON); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-004.

Syntax expression for BP-PROD-005

```
module MK-PHYSICS {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-094 presents the normalized blueprint; Teal TP-268 expresses IF/HOW/THEN; Brown BR-268 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (MK-PHYSICS); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-005.

Syntax expression for BP-PROD-006

```
module JO-RUNTIME {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-006 presents the normalized blueprint; Teal TP-269 expresses IF/HOW/THEN; Brown BR-269 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (JO-RUNTIME); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-006.

Syntax expression for BP-PROD-007

```
module MJOS-SUBSTRATE {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-066 presents the normalized blueprint; Teal TP-270 expresses IF/HOW/THEN; Brown BR-270 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (MJOS-SUBSTRATE); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-007.

Syntax expression for BP-PROD-008

```
module JP-SC-PHONE {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-096 presents the normalized blueprint; Teal TP-271 expresses IF/HOW/THEN; Brown BR-271 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (JP-SC-PHONE); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-008.

Syntax expression for BP-PROD-009

```
module JP-LS-BOAT {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-025 presents the normalized blueprint; Teal TP-272 expresses IF/HOW/THEN; Brown BR-272 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (JP-LS-BOAT); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-009.

Syntax expression for BP-PROD-010

```
module JP-PEACEFUL-PROPULSION {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-010 presents the normalized blueprint; Teal TP-273 expresses IF/HOW/THEN; Brown BR-273 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (JP-PEACEFUL-PROPULSION); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-010.

Syntax expression for BP-PROD-011

```
module SISTER-CONTAINMENT {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-080 presents the normalized blueprint; Teal TP-274 expresses IF/HOW/THEN; Brown BR-274 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (SISTER-CONTAINMENT); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-011.

Syntax expression for BP-PROD-012

```
module EVE-SCREEN {
namespace MJ.PBMEF.Product;
directive evidence_ceiling reference_only;
relationship MJ -> Bangel -> JP -> MK -> JO;
constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-079 presents the normalized blueprint; Teal TP-275 expresses IF/HOW/THEN; Brown BR-275 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (EVE-SCREEN); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-012.

Syntax expression for BP-PROD-013

```
module ANGELS-SAFETY {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-069 presents the normalized blueprint; Teal TP-276 expresses IF/HOW/THEN; Brown BR-276 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (ANGELS-SAFETY); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-013.

Syntax expression for BP-PROD-014

```
module MERMAID-RECON {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-039 presents the normalized blueprint; Teal TP-277 expresses IF/HOW/THEN; Brown BR-277 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (MERMAID-RECON); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-014.

Syntax expression for BP-PROD-015

```
module SHIVA-RAILS {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-077 presents the normalized blueprint; Teal TP-278 expresses IF/HOW/THEN; Brown BR-278 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (SHIVA-RAILS); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-015.

Syntax expression for BP-PROD-016

```
module SHIVAS-INTERNAL {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-069 presents the normalized blueprint; Teal TP-279 expresses IF/HOW/THEN; Brown BR-279 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (SHIVAS-INTERNAL); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-016.

Syntax expression for BP-PROD-017

```
module SHIVA-CRYPTO {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-023 presents the normalized blueprint; Teal TP-280 expresses IF/HOW/THEN; Brown BR-280 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (SHIVA-CRYPTO); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-017.

Syntax expression for BP-PROD-018

```
module SHIVA-EXCHANGE {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-063 presents the normalized blueprint; Teal TP-281 expresses IF/HOW/THEN; Brown BR-281 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (SHIVA-EXCHANGE); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-018.

Syntax expression for BP-PROD-019

```
module MOANA-BROWSER {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-032 presents the normalized blueprint; Teal TP-282 expresses IF/HOW/THEN; Brown BR-282 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (MOANA-BROWSER); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-019.

Syntax expression for BP-PROD-020

```
module AVIATOR-WORKSPACE {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-066 presents the normalized blueprint; Teal TP-283 expresses IF/HOW/THEN; Brown BR-283 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (AVIATOR-WORKSPACE); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-020.

Syntax expression for BP-PROD-021

```
module LECANTO-VISUAL {
namespace MJ.PBMEF.Product;
directive evidence_ceiling reference_only;
relationship MJ -> Bangel -> JP -> MK -> JO;
constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-020 presents the normalized blueprint; Teal TP-284 expresses IF/HOW/THEN; Brown BR-284 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (LECANTO-VISUAL); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-021.

Syntax expression for BP-PROD-022

```
module ELSA-MARKETPLACE {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-048 presents the normalized blueprint; Teal TP-285 expresses IF/HOW/THEN; Brown BR-285 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (ELSA-MARKETPLACE); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-022.

Syntax expression for BP-PROD-023

```
module SHEKINA-CONTINUITY {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-019 presents the normalized blueprint; Teal TP-286 expresses IF/HOW/THEN; Brown BR-286 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (SHEKINA-CONTINUITY); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-023.

Syntax expression for BP-PROD-024

```
module PEARL-CONTAINMENT {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-035 presents the normalized blueprint; Teal TP-287 expresses IF/HOW/THEN; Brown BR-287 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (PEARL-CONTAINMENT); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-024.

Syntax expression for BP-PROD-025

```
module Bangel -SPREADSHEETS {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-002 presents the normalized blueprint; Teal TP-288 expresses IF/HOW/THEN; Brown BR-288 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (Bangel -SPREADSHEETS); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-025.

Syntax expression for BP-PROD-026

```
module JPOSMJ-PORTFOLIO {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-076 presents the normalized blueprint; Teal TP-289 expresses IF/HOW/THEN; Brown BR-289 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (JPOSMJ-PORTFOLIO); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-026.

Syntax expression for BP-PROD-027

```
module STANDARD-CONTAINMENT-GEOMETRY {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-002 presents the normalized blueprint; Teal TP-290 expresses IF/HOW/THEN; Brown BR-290 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (STANDARD-CONTAINMENT-GEOMETRY); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-027.

Syntax expression for BP-PROD-028

```
module RESERVOIR-COLLECTION-MODULATION {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-059 presents the normalized blueprint; Teal TP-291 expresses IF/HOW/THEN; Brown BR-291 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (RESERVOIR-COLLECTION-MODULATION); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-028.

Syntax expression for BP-PROD-029

```
module MJOS-SOLAR-CORE-001 {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-084 presents the normalized blueprint; Teal TP-292 expresses IF/HOW/THEN; Brown BR-292 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (MJOS-SOLAR-CORE-001); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-029.

Syntax expression for BP-PROD-030

```
module MJOS-SOLAR-SEMI-PHONE-001 {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-049 presents the normalized blueprint; Teal TP-293 expresses IF/HOW/THEN; Brown BR-293 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (MJOS-SOLAR-SEMI-PHONE-001); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-030.

Syntax expression for BP-PROD-031

```
module MJOS-SOLAR-MARINE-001 {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-064 presents the normalized blueprint; Teal TP-294 expresses IF/HOW/THEN; Brown BR-294 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (MJOS-SOLAR-MARINE-001); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-031.

Syntax expression for BP-PROD-032

```
module MJOS-SOLAR-ROCKET-001 {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-092 presents the normalized blueprint; Teal TP-295 expresses IF/HOW/THEN; Brown BR-295 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (MJOS-SOLAR-ROCKET-001); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-032.

Syntax expression for BP-PROD-033

```
module DRO-COLLECTION-001 {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-033 presents the normalized blueprint; Teal TP-296 expresses IF/HOW/THEN; Brown BR-296 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (DRO-COLLECTION-001); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-033.

Syntax expression for BP-PROD-034

```
module DRO-PEARL-HANDOFF-001 {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-090 presents the normalized blueprint; Teal TP-297 expresses IF/HOW/THEN; Brown BR-297 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (DRO-PEARL-HANDOFF-001); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-034.

Syntax expression for BP-PROD-035

```
module DRO-IGNITION-REFERENCE-001 {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-068 presents the normalized blueprint; Teal TP-298 expresses IF/HOW/THEN; Brown BR-298 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (DRO-IGNITION-REFERENCE-001); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-035.

Syntax expression for BP-PROD-036

```
module LECANTO-IMAGE-TRUTH-001 {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-098 presents the normalized blueprint; Teal TP-299 expresses IF/HOW/THEN; Brown BR-299 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (LECANTO-IMAGE-TRUTH-001); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-036.

Syntax expression for BP-PROD-037

```
module JP-SC-DRO-PEARL-INTEGRATION-001 {
  namespace MJ.PBMEF.Product;
  directive evidence_ceiling reference_only;
  relationship MJ -> Bangel -> JP -> MK -> JO;
  constraint permit = MJ.required;
}
```

EXACT MATH OR TYPED IDENTITY

No direct formula; typed representation and evidence pathway apply.

CODE -> MATH -> IMAGE -> PROCESS FLOW

Syntax is parsed as a reference candidate; formula or identity resolves to JP; MK classifies physics applicability; JO verifies; visual reference YP-010 presents the normalized blueprint; Teal TP-300 expresses IF/HOW/THEN; Brown BR-300 expresses the R lifecycle; MJ remains the permit boundary.

RELATED WHITE/PINK SOURCES

PROJECT-PRODUCT-REGISTER.json (JP-SC-DRO-PEARL-INTEGRATION-001); product White Paper; formulas and objects referenced by the product; passkey scope; Blue Page BP-PROD-037.