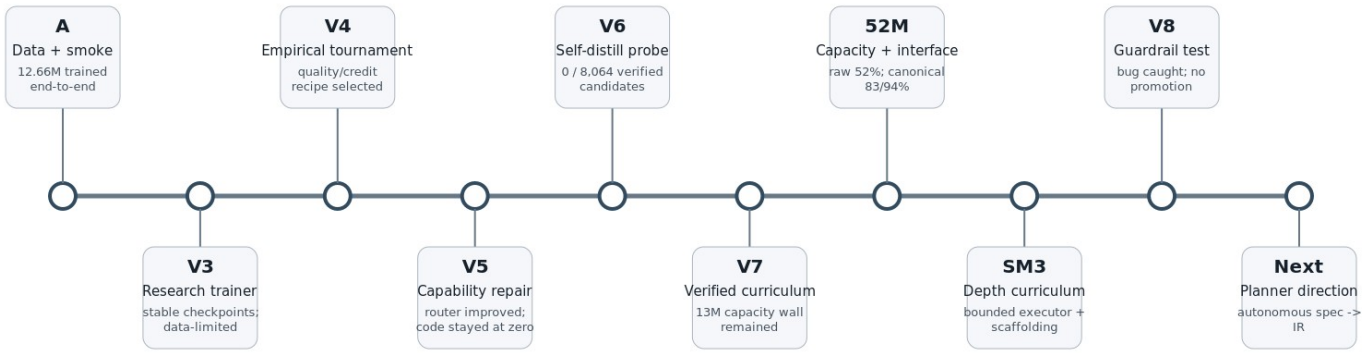
VIETNAMESE SMALL CODE MODEL RESEARCH PREVIEW

Từ pipeline 13M đến executor 52M có kiểm chứng

TDGL Research - 13 July 2026



Publication-safe summary. Full recipes and checkpoints remain private.

1. Executive summary / Tóm tắt

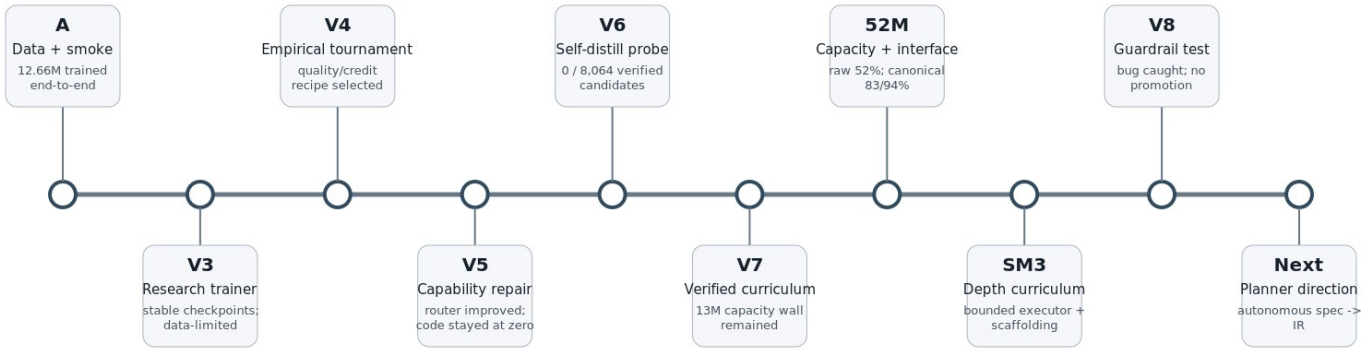
Dự án xây một mô hình code nhỏ ưu tiên tiếng Việt từ random initialization, với tiêu chí đánh giá bằng thực thi chứ không chỉ loss. Chuỗi thí nghiệm đã xác định bốn đòn bẩy chính: capacity, interface, structural/depth coverage và verifier-guided training.

The project trained a Vietnamese-first small code model from scratch and evaluated it with executable tests. The research identified four principal levers: model capacity, deterministic interfaces, structural/depth coverage, and verification-guided training.

Model 13M chạy ổn định nhưng không vượt executable pass@1=0. Một probe 52M đạt 52% raw pass@1; canonical signatures cộng AST binding tăng lên khoảng 83% test và 94% identifier robustness trong synthetic grammar.

The strongest results remain synthetic and scoped. Chunked-depth 100% used a correct operation sequence supplied by the harness and must not be described as autonomous planning.

2. Project timeline



Phase	Status	Core result
Phase A	Validated	End-to-end data/tokenizer/train/SFT/stage pipeline.
V3.2	Validated	13M trainer stable; validation improved; data budget low.
V4.1	Validated	Empirical recipe selection; code pass@1 stayed zero.
V5	Validated negative	Router improved; code capability stayed zero.
V6	Validated negative	0 verified solutions from 8,064 candidates.
V7	Validated negative	260-family curriculum did not lift 13M pass@1.
52M/SM3	Validated in scope	Capacity + canonical interface produced strong synthetic results.
V8	Invalidated	Schema bug dropped task specifications; no promotion.
Next	Designed	Autonomous specification-to-IR planning.

3. Quantitative milestones

Milestone	Model	Selected measurement	Interpretation
A-to-Z smoke	12.66M	Pretrain 9.7476 -> 5.8610; SFT 6.6090 -> 2.2739	Pipeline proof, not capability proof.
V3.2	12.72M	Validation 5.8361 -> 5.1485	Technically stable, data-constrained.
V4.1	12.72M	5.0340 -> 4.8261; pass@1=0	Loss improved without executable capability.
V5	12.72M	Router macro-F1 0.8958; code=0	Policy separable from code skill.
V6	12.72M	0/8,064 verified samples	Self-distillation too early.
V7	12.72M	260 families; code=0	Practical 13M capacity wall.
Raw capacity probe	~52.6M	52% pass@1 on n=100 synthetic sample	Capacity mattered.
Canonical + AST	~52.6M	83% test; 94% robustness; 100% syntax	Interface mattered as much as scale.
Structural curriculum	~52.6M	0/6 -> 5/6; held-out ~99%	Program structure must be covered.

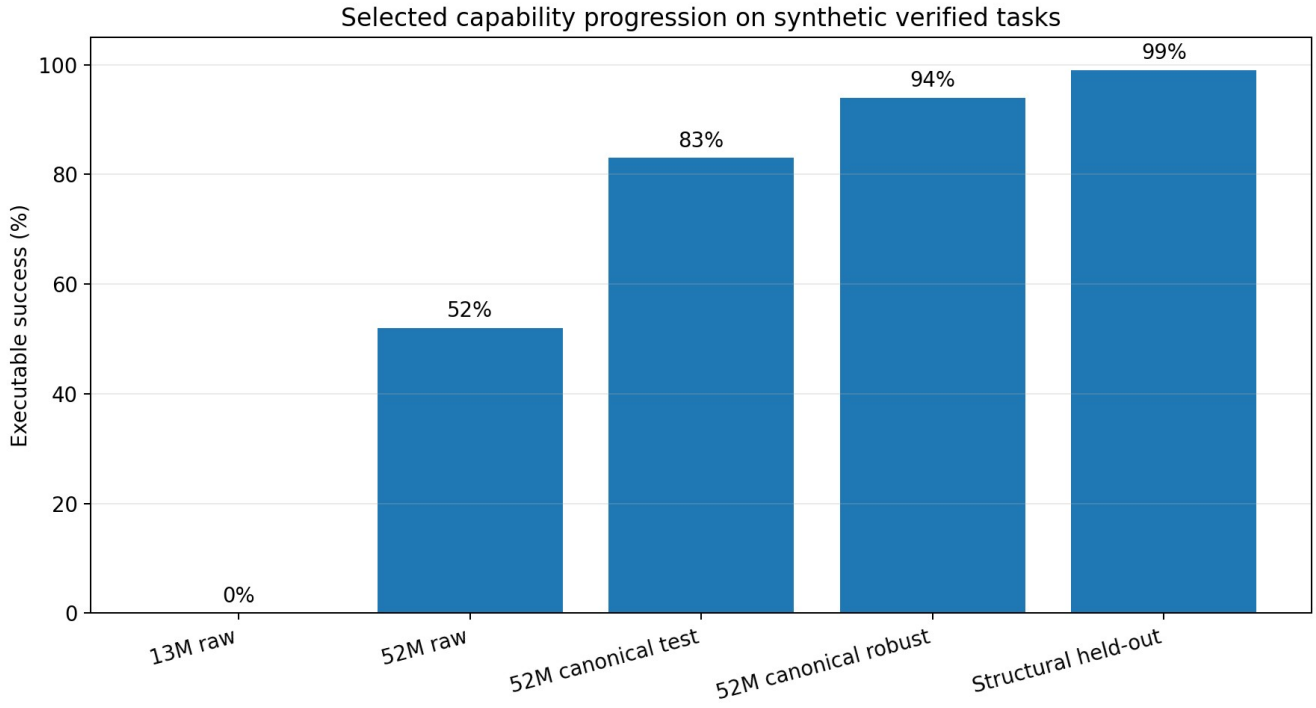


Figure 1. Selected synthetic capability progression. Metrics are not directly comparable to public coding benchmarks.

4. What actually moved capability

Capacity

The 13M model remained at zero executable pass@1 even after a verified curriculum. A 52M probe crossed the wall, establishing a practical minimum scale for the current data family.

Interface

Canonical signatures and deterministic AST rebinding removed arbitrary identifier copying from the model task. This produced the largest single capability jump.

Structure and depth

Adding more values inside one template did not teach new program structures. Explicitly varying order, repeated filters/maps, and depth expanded the useful region.

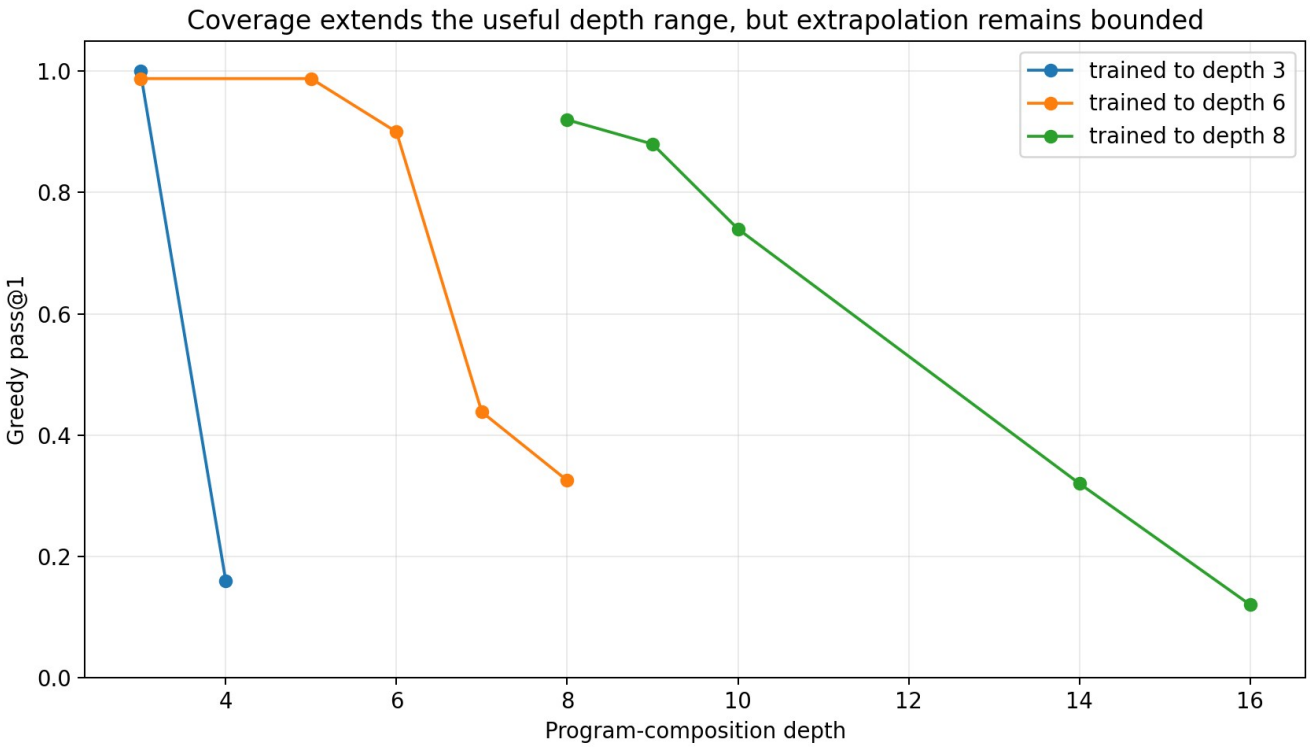
Verification

AST checks, sandboxed assertions, validation-only promotion, and lineage hashes prevented invalid candidates from replacing working checkpoints.

Self-training

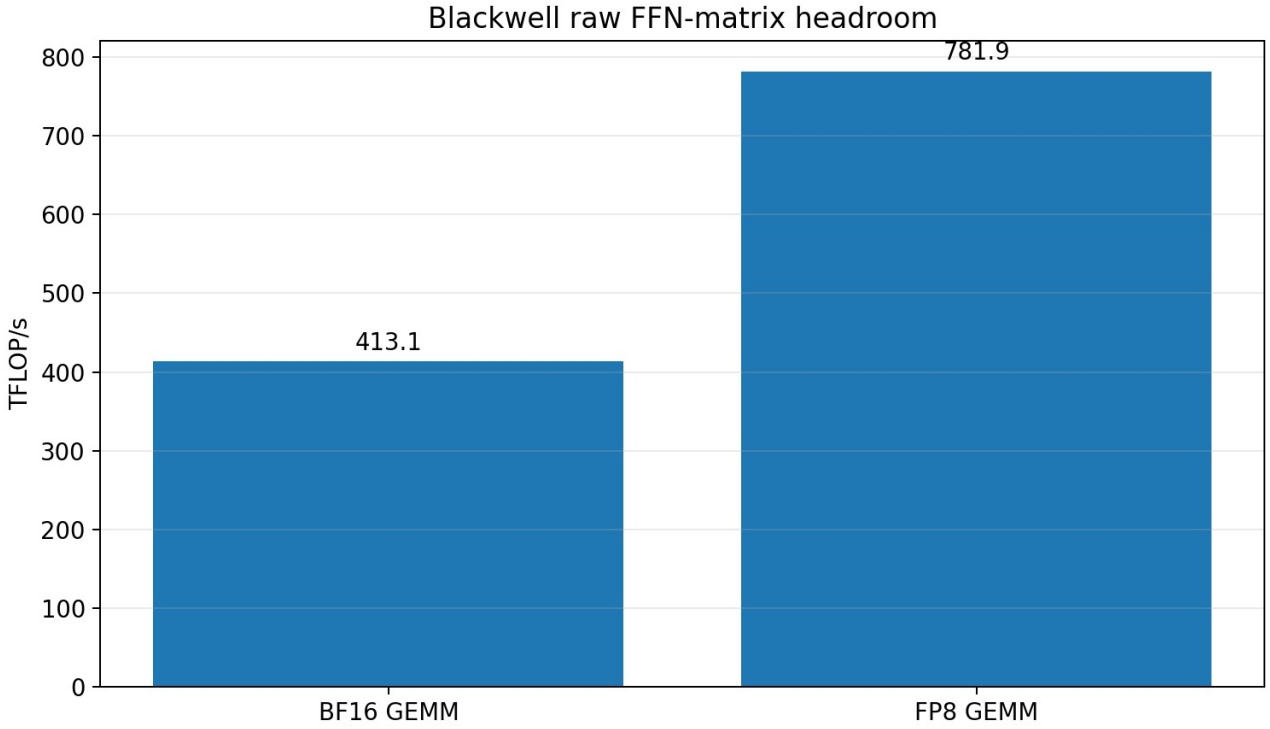
Verified replay improved greedy performance only when correct sampled trajectories existed. It stopped helping once pass@1 reached pass@32.

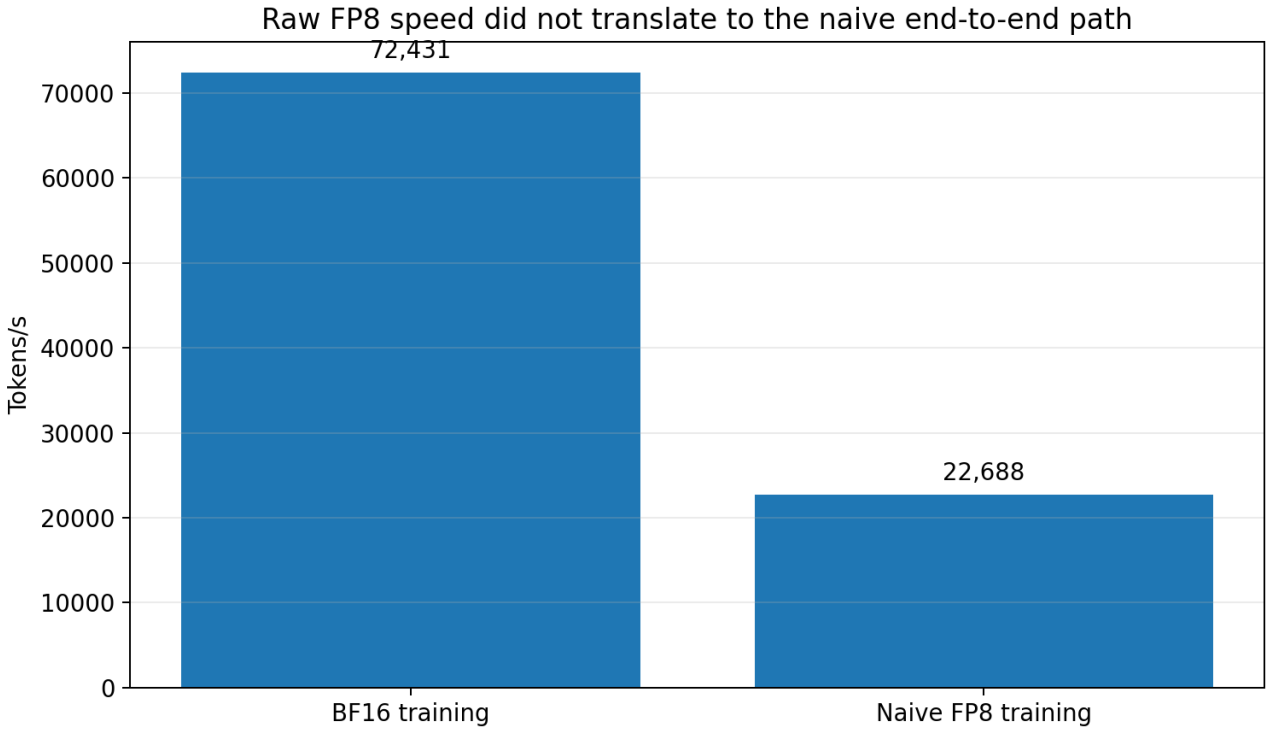
5. Depth generalization



Coverage pushed the boundary outward and softened the drop, but did not create unlimited extrapolation. Reported points come from separate synthetic evaluation batches and should be treated as representative rather than a single unified benchmark run.

6. Hardware findings





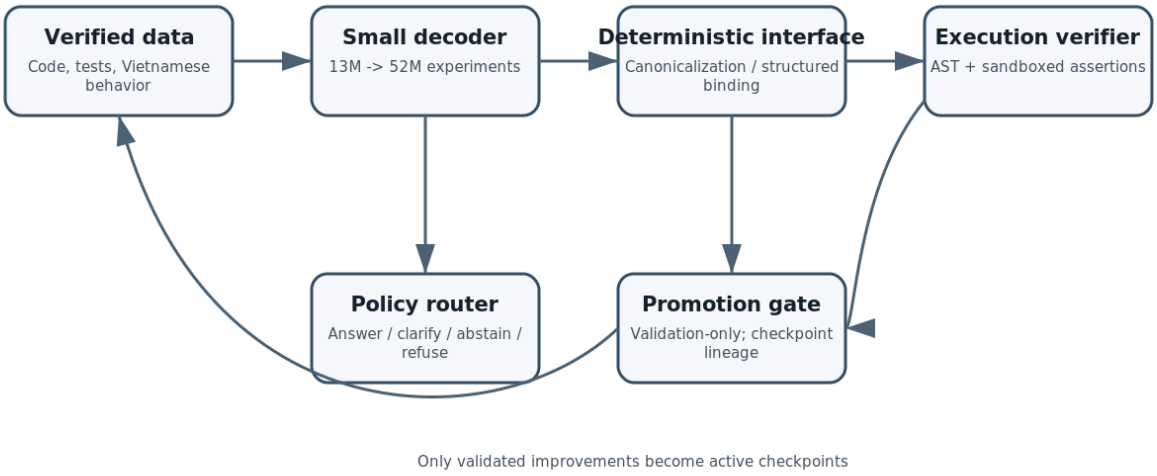
Measurement	Result	Meaning
Raw BF16 FFN GEMM	413.1 TFLOP/s	Approximate matrix-kernel ceiling.
Raw FP8 FFN GEMM	781.9 TFLOP/s	Large theoretical headroom.
Peak sustained transformer	224.7 TFLOP/s, MFU 0.547	Measured at ~1.445B parameters.
BF16 end-to-end ~0.44B	72,431 token/s	Fast, stable baseline.
Naive FP8 end-to-end ~0.44B	22,688 token/s	Custom unfused FP8 path lost to overhead.

7. Reliability and invalidated evidence

The official V8 run silently dropped task descriptions because a transformation expected a field that did not exist in the stored JSONL schema. The resulting near-zero metrics are invalid as capability evidence. The candidate was not promoted, demonstrating the value of a validation gate and independent probe.

A separate inference decomposition reached 100% at much larger synthetic depths only because the harness already possessed the correct operation list. The proper description is oracle decomposition, not autonomous reasoning.

8. System direction



The emerging system is not only a decoder. It combines a small model, policy router, deterministic interface, execution verifier, persistent checkpoints, and validation-only promotion. The next missing component is an autonomous specification-to-IR planner.

9. Public claims and limitations

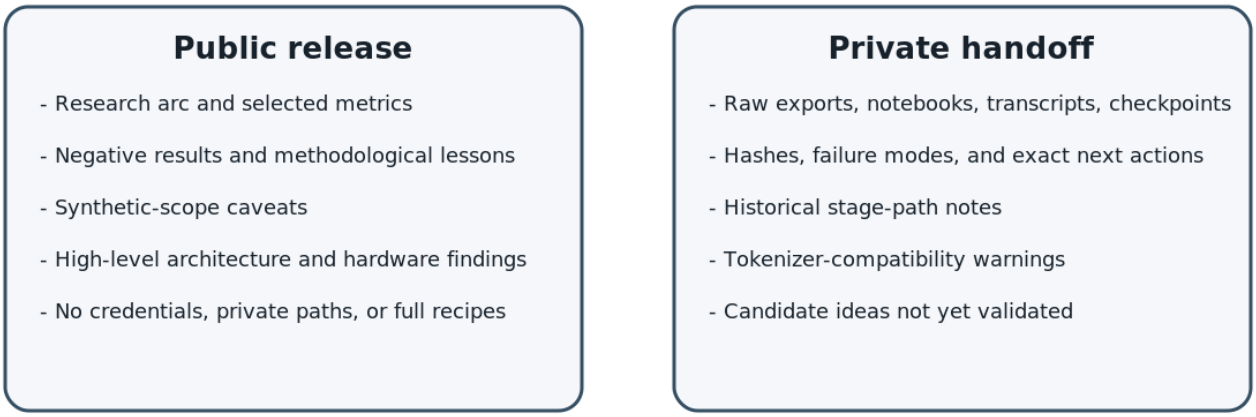
Supported statement	Required caveat
52M crossed the 13M practical wall	On the project synthetic verified task family.
Canonical interface improved robustness	Inside a deterministic signature/AST wrapper.
Depth coverage expands generalization	Extrapolation remains bounded.
STaR improved greedy capability	Only until pass@1 reached pass@k.
Chunking solved very deep tasks	The harness supplied the correct decomposition.
Blackwell FP8 has raw headroom	Naive end-to-end FP8 was slower in the tested stack.

10. Next research target

Train and evaluate a planner that receives only a Vietnamese specification and function signature, emits a bounded IR, and relies on a deterministic compiler and hidden execution tests.

Compare it against direct code generation under the same parameter and token budgets. This removes the largest remaining oracle assumption.

Appendix A - Public/private boundary



The public release includes selected metrics, diagrams, articles, and explicit limitations. The private handoff contains raw notebook exports, transcripts, checkpoints, hashes, and continuation instructions. No credential is stored in either package.

Appendix B - Artifact map

Public file	Purpose
README.md	Short bilingual GitHub overview.
tdgl_codes_article_vi.md / .html	Long Vietnamese website article.
tdgl_codes_article_en.md / .html	English research preview.
figures/	SVG and PNG charts/diagrams.
data/public_metrics.csv	Selected publication-safe metrics.
PUBLICATION_BOUNDARY.md	What to publish and what to retain.