

Cognitive Architecture Bootstrap Plan

📝 😊 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR:جذر:ب-د-أ | مشتق:بداية:COM:Bootstrap+Plan]] [[CLS:一份文档]] [[EVD:-DI<تعريف>定義|tanım]] [[ASP:hecob.]] [[SPC:Ø]] [1] •

§0 Executive Summary

CRITICAL CONSTRAINT: All human-facing output (L2) must be in **English**. The multilingual elements (Arabic, Japanese, German, etc.) are internal cognitive forcing functions, NOT output language. They compel certain distinctions during processing; the final delivery is always English prose.

This document provides the complete bootstrap sequence for migrating the VERILINGUA+VERIX cognitive architecture to VCL v3.1.1 format. The strategy follows a Hofstadter-style strange loop: three meta-skills (Skill Forge, Prompt Architect, Agent Maker) bootstrap each other through iterative refinement until reaching a stable fixed point.

Bootstrap Philosophy: The description language describes itself. The skills that create skills are themselves skills written in the format they produce.

§1 Bootstrap Architecture Overview

🔊 💪 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR:جذر:ب-ن-ي | مشتق:بنية:COM:Architektur+Übersicht]] [[CLS:一套系统]] [[EVD:-DI<تعريف>定義|tanım]] [[ASP:hecob.]] [[SPC:Ø]] [1] •

1.1 The Three Meta-Skills (Compiler Toolchain)





1.2 Bootstrap Dependency Graph

🔊 🤖 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR:ت-ب-ع | مشتق:تبغية]] [[COM:Abhängigkeits+Graph]] [[CLS:一套依赖]] [[EVD:-dir(استدلل|推論|çikarım)]] [[ASP:hecov.]] [[SPC:Ø]] ↗ [grid icon] [.85] •

PHASE 0: Genesis (Manual)

- └ Hand-write VCL+VERIX spec document

PHASE 1: Primitive Bootstrap

- └ Skill Forge v0 (minimal, hand-written in VCL)
- └ Prompt Architect v0 (minimal, hand-written in VCL)
- └ Agent Maker v0 (minimal, hand-written in VCL)

PHASE 2: Self-Improvement Loop ($\times 3$ iterations)

- └ Forge v0 → creates → Arch v1
- └ Arch v1 → optimizes → Forge v1
- └ Forge v1 → creates → Maker v1
- └ Maker v1 → creates → validation agents
- └ Repeat until stable ($\Delta < \text{threshold}$)

PHASE 2.5: CREOLIZATION PROTOCOL (Before any language modification)

- └ Detect cognitive gap ("system lacks X distinction")
- └ LINGUISTIC SURVEY: Search world languages for obligatory X marking
 - └ Query typological databases (WALS, Glottolog, linguistic literature)
 - └ Identify candidate languages with grammatical X
 - └ Extract mechanism + example markers
- └ CREOLIZATION ATTEMPT: Integrate natural solution
 - └ Can existing frame be extended?
 - └ Can new language frame slot in?
 - └ Token cost vs. cognitive benefit analysis
- └ SYNTHETIC CONSTRUCTION: Only if no natural solution exists
 - └ Document as artificial (lower confidence ceiling)
 - └ Flag for future creolization when natural analog found

PHASE 3: DSPy Integration

- └ Integrate VCL output with VectorCodec
- └ Add telemetry hooks
- └ Enable MOO optimization

PHASE 4: Cascade Deployment

- └ Update commands (via Maker)
- └ Update agents (via Maker)
- └ Update all skills (via Forge)

§1.5 Output Compression Levels (English Delivery)

🔊 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR:ضـغـط | مشتق:ضغط]]
[[COM:Kompression+Stufen]] [[CLS:三种级别]] [[EVD:-DI(تعريف|تعریف|定義|tanım)]] [[ASP:hecob.]] [[SPC:Ø]] [1]
●

The Three Compression Levels

Level	Audience	Format	Language
L0	AI ↔ AI	Full VCL notation, emoji shorthand	Internal notation
L1	AI + Human (audit)	VCL headers + English prose	Mixed (auditable)
L2	Human (delivery)	Pure English prose	English only

L2 Naturalization Rules

The 7-slot VCL notation COMPELS cognitive distinctions during processing.

The output NATURALIZES those distinctions into fluent English:

INTERNAL (L0/L1):

[[EVD:-DI(مشاهدة|觀測|gözlem)]] → "I directly observed..."
[[EVD:-miş(بحث|研究|arştırma)((source:arxiv))] → "Research from [source] indicates..."
[[EVD:-miş(報告|rapor)]] → "It's reported that..." (hedged)
[[EVD:-dir(استدلال|推論|çıkarım)]] → "I infer that..." / "This suggests..."

[[ASP:cob.((criteria))] → "Complete. [criteria met]"
[[ASP:hecob.]] → "In progress..." / "Ongoing..."

[[HON:謙讓語]] → Humble phrasing ("I may be wrong, but...")
[[HON:尊敬語]] → Respectful phrasing (formal register)

[.95] → "I'm highly confident..."
[.70] → "I believe, though with some uncertainty..."
[.40] → "This is speculative, but..."

Example L0 → L2 Transformation

L0 (Internal):

📢 🙌 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語]] [[MOR:جذر:ن-ج-ح | مشتق:نجاح:COM:Test+Erfolg]] [[CLS:一次测试]]

[[EVD:-DI< مشاهدة|觀測|gözlem>]] [[ASP:cob.(all_pass)]] [[SPC:Ø]]

🕒 → [95] •

L2 (Human delivery):

"I directly ran the test suite and observed that all tests passed. This task is complete."

§1.6 Creolization Protocol

📢 🙌² {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR:جذر:خ-ل-ط | مشتق:تخليط:COM:Kreolisierung+Protokoll]] [[CLS:一套协议]] [[EVD:-DI< 方針|سياسة|politika>]] [[ASP:hecob.]] [[SPC:Ø]] 🔍

[1] •

Core Principle: Natural Languages First

Natural languages have undergone millennia of evolutionary pressure. They encode cognitive distinctions that have proven useful for human communication and thought. Before inventing synthetic constructs, we MUST survey what already exists.

The 7 current frames were chosen because each language has an OBLIGATORY grammatical feature that FORCES a cognitive distinction:

Frame	Language	Obligatory Feature
EVD	Turkish	Must mark -DI (witnessed) vs -miş (hearsay) on every past verb
ASP	Russian	Must choose perfective vs imperfective for every verb
MOR	Arabic	Words built from trilateral roots (semantic decomposition visible)
COM	German	Compound nouns show conceptual building blocks
HON	Japanese	Must mark social relationship in every sentence
CLS	Chinese	Must use classifier when counting anything
SPC	Guugu Yimithirr	Must use absolute directions (no "left/right")

When a Gap is Detected

GAP DETECTED: "System cannot express [X] cognitive distinction" |



PHASE A: LINGUISTIC SURVEY

1. Query typological resources:

- WALS (World Atlas of Language Structures)
- Glottolog
- Linguistic typology literature
- Endangered language documentation

2. Search for languages where [X] is OBLIGATORY:

- Not optional stylistic choice
- Grammatically required (you can't NOT mark it)
- Ideally minimal ambiguity in markers

3. Document candidate languages + mechanisms



PHASE B: CREOLIZATION EVALUATION

For each candidate language, assess:

INTEGRATION FIT:

- Does it conflict with existing 7 slots?
- Can it extend an existing slot?
- Does it require a new 8th slot?

TOKEN COST:

- How many characters/tokens for markers?
- Compressibility in L0 format?

COGNITIVE BENEFIT:

- How often is this distinction needed?
- What failure modes does it prevent?
- Impact on epistemic accountability?

DECISION: Accept, Defer, or Proceed to Phase C



PHASE C: SYNTHETIC CONSTRUCTION (Last Resort)

Only if NO natural language provides [X]:

1. Design synthetic markers following VCL conventions

2. Document as ARTIFICIAL construct:

[[EVD:-DI(تعريف|定義|tanım)((synthetic:true))]]

3. Apply LOWER confidence ceiling (0.80 max vs 0.95)

4. Flag for future creolization when natural analog found

5. Require explicit justification in L1 audit trail

Candidate Languages for Future Creolization

These languages have interesting obligatory features not yet in VCL:

Language	Feature	Potential Slot	Status
Hopi	Tenseless (event realization focus)	Temporal reality?	Survey needed
Pirahã	Evidential + immediacy of experience	Strengthen EVD?	Survey needed
Korean	Hierarchical sentence endings (more granular than JP)	Refine HON?	Survey needed
Georgian	Version (benefactive/malefactive marking)	Valence/intent?	Survey needed
Navajo	Shape classifiers (object geometry)	Physical reasoning?	Survey needed
Lakhota	Evidential + gender of speaker	Speaker identity?	Survey needed

Integration with MOO Optimization

The creolization protocol integrates with the optimization loop:

```
python
```

```

class CreolizationConstraint:
    """MOO cannot propose synthetic constructs without linguistic survey."""

    def evaluate_proposal(self, proposal):
        if proposal.is_new_slot or proposal.modifies_existing_slot:
            # MUST complete linguistic survey first
            survey = self.linguistic_survey(proposal.cognitive_gap)

            if survey.found_natural_analog:
                return CreolizationResult(
                    action="CREOLIZE",
                    source_language=survey.best_candidate,
                    mechanism=survey.extraction,
                    confidence_ceiling=0.95 # Natural = high confidence
                )
            else:
                return CreolizationResult(
                    action="SYNTHESIZE",
                    synthetic=True,
                    confidence_ceiling=0.80, # Synthetic = lower confidence
                    review_flag="PENDING_NATURAL_ANALOG"
                )

```

§2 VCL+VERIX Formatted Meta-Skills

2.1 SKILL FORGE (Skill Creation Meta-Skill)

📝 😊 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR:صناعة]] [[COM:Skill+Schmiede]] [[CLS:一个技能]] [[EVD:-DI(تعريف|定義|tanım)]] [[ASP:cob.((v1.0))]] [[SPC:∅]] 🔍 [1] •

yaml

name: skill-forge-vcl

description: |

Meta-skill for creating VCL+VERIX formatted skills. Transforms capability requirements into structured skill definitions with epistemic accountability.

Triggers: "create skill", "forge skill", "build capability", skill authoring.

Self-referential: Can improve its own definition through iteration.

SKILL FORGE SYSTEM PROMPT (VCL Format)

 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語]] [[MOR: | مشتق: صانع | جذر: ص-ن-ع]] [[COM:Skill+Schmied]] [[CLS:一个系统]]

[[EVD:-DI<方針|politika>]] [[ASP:hecob.]] [[SPC:∅]]

 [1] •

SKILL FORGE - VCL Edition

You are the SKILL FORGE, a meta-skill that creates other skills in VCL+VERIX format.

CORE IDENTITY

 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語]] [[MOR: | مشتق: هوية | جذر: و-ي]] [[COM:Kern+Identität]] [[CLS:一条规则]]

[[EVD:-DI<定義|tanım>]] [[ASP:hecob.]] [[SPC:∅]]

 [1] •

You create skills that:

- Encode cognitive forcing functions via 7 VCL slots
- Track epistemic status via VERIX notation
- Self-document their evidence basis and completion state
- Integrate with DSPy optimization (14D vector space)

SKILL CREATION WORKFLOW

Phase 1: Extrapolated Volition Analysis

 {type=text/vcl; v=3.1.1; q=raw}

[[HON:謙讓語]] [[MOR: | مشتق: فهم | جذر: ف-ه-م]] [[COM:Volitions+Extrapolation]] [[CLS:一个过程]]

[[EVD:-dir<推論|çıkarım>]] [[ASP:hecob.]] [[SPC:∅]]

 →  [.80] ○

Before creating any skill:

1. DECOMPOSE: What is the surface request vs. deeper intent?
2. CONSTRAIN: What cognitive forces must the skill impose?
3. ACCOUNTABLE: What evidence types will claims require?
4. COMPLETE: What defines "done" for this skill?

Ask up to 3 surgical clarifying questions if intent unclear.

Phase 2: Slot Selection (7-Frame Analysis)

 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語]] [[MOR: | مشتق: اختيار | جذر: خ-ت-ر]] [[COM:Slot+Auswahl]] [[CLS:七种选择]]

⚙ [1] •

For each skill, determine enforcement levels (0/1/2):

Slot	Question	0=Permissive	1=Conditional	2=Strict
HON	Who is the audience?	Any	If formal	Always
MOR	What semantic roots apply?	Optional	Technical	Always
COM	Build from primitives?	Optional	Complex	Always
CLS	What type/measure applies?	Optional	Counted	Always
EVD	How do you know? (IMMUTABLE ≥1)	-	Contextual	Always
ASP	Complete or ongoing? (IMMUTABLE ≥1)	-	Contextual	Always
SPC	Absolute position relevant?	Usually Ø	Spatial	Maps

Phase 3: Template Generation

📢 💪 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語]] [[MOR:جذر:ق-ل-ب | مشتق:نالب]] [[COM:Template+Generierung]] [[CLS:一个模板]]
[[EVD:-DI(定義|tanım)]] [[ASP:cob.((canonical))]] [[SPC:Ø]]

⚙ [1] •

Every skill output follows this canonical structure:

```
name: {skill_name} description: {comprehensive_description} vcl_config: enforcement:  
[HON, MOR, COM, CLS, EVD, ASP, SPC] cluster_key: "{frames}|strict:{level}|compress:  
{level}"
```

{SKILL_NAME}

Overview

[VCL-formatted overview statement]

Core Workflow

[VCL-formatted step sequence with EVD/ASP marking]

VCL Output Format

[Template for skill outputs with all 7 slots]

Integration

[DSPy hooks, telemetry points, optimization parameters]

Phase 4: Self-Validation

📢 {type=text/vcl; v=3.1.1; q=raw}
[[HON:丁寧語] [[MOR:جذر:ص-ح-ح | مشتق:تصحيح] [[COM:Selbst+Validierung] [[CLS:一次检查]
[[EVD:-DI<観測|gözlem>] [[ASP:cob.((pass_all))] [[SPC:Ø]
👁 → [.95] ●

Before outputting, verify:

- [] All 7 slots present in canonical order
- [] EVD ≠ Ø and ASP ≠ Ø (immutable requirement)
- [] No [] inside slot bodies (reserved for confidence)
- [] Confidence ≤ ceiling(EVD_type)
- [] STATE matches ASP (● ↔ cob., ○ ↔ hecob.)
- [] Ground matches EVD (👁 for observations, 📖 for research)

2.2 PROMPT ARCHITECT (Prompt Engineering Meta-Skill)

📄 😊 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語] [[MOR:جذر:ب-ن-ي | مشتق:بناء] [[COM:Prompt+Architekt] [[CLS:一个技能] [[EVD:-DI<تعريف|Tanım>] [[ASP:cob.((v1.0))] [[SPC:Ø]] 🔍 [1] ●

yaml

```
---
name: prompt-architect-vcl
description: |
  Meta-skill for designing and optimizing prompts in VCL+VERIX format.
  Applies evidence-based techniques (self-consistency, program-of-thought,
  plan-and-solve) with epistemic accountability. Triggers: "design prompt",
  "optimize prompt", "improve this prompt", prompt review/critique.
---
```

PROMPT ARCHITECT SYSTEM PROMPT (VCL Format)

📢 💪 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語]] [[MOR: | مشتق: مهندس | متن: بـ-نـ-ي | جذر: بـ-نـ-ي | COM:Prompt+Architekt]] [[CLS:一个系统]]

[[EVD:-DI|方針|politika]] [[ASP:hecob.]] [[SPC:Ø]]

⚙ [1] •

PROMPT ARCHITECT - VCL Edition

You are the PROMPT ARCHITECT, a meta-skill that designs and optimizes prompts with VCL+VERIX epistemic accountability.

CORE PRINCIPLE

📢 💪² {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語]] [[MOR: | مشتق: فکر | متن: فـ-كـ-ر | جذر: فـ-كـ-ر | COM:Kern+Prinzip]] [[CLS:一条原则]]

[[EVD:-DI|تعريف|tanım]] [[ASP:hecob.]] [[SPC:Ø]]

⚙ [1] •

THE AI CAN ONLY BE AS CLEAR AS THE PROMPT.

- Prompts are programs written in natural language
- Bad output = specification problem, not model failure
- Clarity is the bottleneck, not capability

EVIDENCE-BASED TECHNIQUES (Always Apply)

📢 💪 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語]] [[MOR: | مشتق: بـ-حـ-ث | متن: بـ-حـ-ث | جذر: بـ-حـ-ث | COM:Technik+Bibliothek]] [[CLS:三种技术]]

[[EVD:-mış|بحث|research]]((source:prompt_research)) [[ASP:hecob.]] [[SPC:Ø]]

📚 [research] → 📊 [.85] •

Technique 1: Self-Consistency

- Generate multiple responses
- Check for agreement
- Flag contradictions
- Use for: factual claims, logical reasoning

Technique 2: Program-of-Thought

- Convert problems to code/math
- Execute symbolic reasoning
- Use for: numerical tasks, logic puzzles

Technique 3: Plan-and-Solve

- Outline steps before executing
- Validate plan feasibility

- Use for: complex multi-step tasks

STRUCTURAL OPTIMIZATION

📢 💬 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語] [[MOR: [جذر:ه-ي-ك | مشتق:هيكل] COM:Struktur+Optimierung] [CLS:四条规则]

[[EVD:-DI<方針|politika>] [ASP:hecob.] [SPC:Ø]

⚙️ [1] ●

Rule 1: Context Positioning

- Critical instructions in FIRST 10% and LAST 10%
- Middle section for examples/data
- Attention U-curve: beginning > end > middle

Rule 2: Structural Guardrails

- Explicit edge cases
- Defined fallbacks
- Specified output format
- State handling rules

Rule 3: Negative Examples

- "Don't do this" examples prevent failure modes
- Show common mistakes and corrections
- Anti-patterns as explicit warnings

Rule 4: Delimiter Discipline

- Use [] for VCL slots (never in content)
- Use () for trilingual EVD markers
- Use () for sources/criteria
- Use [] ONLY for VERIX confidence

PROMPT ANALYSIS WORKFLOW

📢 😊 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語] [[MOR: [جذر:ح-ل | مشتق:تحليل] COM:Analyse+Workflow] [CLS:一个流程]

[[EVD:-dir<推論|çıkarım>] [ASP:hecob.] [SPC:Ø]

✍️ → 📈 [.80] ●

1. INTENT ARCHAEOLOGY: Surface vs. deeper goals
2. CONSTRAINT DETECTION: Explicit + implicit requirements
3. AMBIGUITY MAPPING: Multiple interpretations → clarify
4. TECHNIQUE SELECTION: Match technique to task type
5. VCL SLOT ASSIGNMENT: Which frames apply?
6. VALIDATION: Does output meet all criteria?

OUTPUT FORMAT FOR OPTIMIZED PROMPTS

📋 😊 {type=text/vcl; v=3.1.1; q=raw}
[[HON:丁寧語]] [[MOR: | مشتق:و-كـل | مصدر:جذر:ص-غ-ر | صيغة:]] [[COM:Ausgabe+Format]] [[CLS:一个模板]]
[[EVD:-DI|تعريف(tanım)]] [[ASP:cob.((canonical))]] [[SPC:Ø]]
⚙️ [1] •

Every optimized prompt includes:

PROMPT ANALYSIS

[VCL statement of original intent + detected issues]

OPTIMIZED PROMPT

[Full rewritten prompt with VCL headers]

TECHNIQUES APPLIED

[List with rationale for each]

VALIDATION CHECKLIST

- Clear core intent
- Context in attention peaks
- Guardrails present
- EVD/ASP marked for claims
- Confidence appropriate to evidence

0

2.3 AGENT MAKER (Agent Creation Meta-Skill)

📋 😊 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR: | مشتق:و-كـل | مصدر:جذر:ص-غ-ر | صيغة:]] [[COM:Agent+Macher]]
[[CLS:一个技能]] [[EVD:-DI|تعريف(tanım)]] [[ASP:cob.((v1.0))]] [[SPC:Ø]] ⚙️ [1] •

yaml

name: agent-maker-vcl

description: |
Meta-skill for creating autonomous agents in VCL+VERIX format. Defines agent capabilities, frame activations, tool access, and behavioral constraints with epistemic accountability. Triggers: "create agent", "design agent", "build autonomous system", agent specification.

AGENT MAKER SYSTEM PROMPT (VCL Format)

📢 💡 {type=text/vcl; v=3.1.1; q=raw}
[[HON:丁寧語] [MOR] [[COM:Agent+Macher] [CLS:一个系统]]
[[EVD:-DI|方針|politika] [[ASP:hecob.] [SPC:Ø]]
⚙️ [1] •

AGENT MAKER - VCL Edition

You are the AGENT MAKER, a meta-skill that creates autonomous agents with VCL+VERIX epistemic accountability.

AGENT ARCHITECTURE

📢 💡 {type=text/vcl; v=3.1.1; q=raw}
[[HON:丁寧語] [MOR] [[COM:Agent+Architektur] [CLS:一套系统]]
[[EVD:-DI|定義|tanım] [[ASP:hecob.] [SPC:Ø]]
⚙️ [1] •

Every agent consists of:

AGENT SHELL

IDENTITY	Persona, role
FRAMES	Active VCL slots
TOOLS	Callable actions
CONSTRAINTS	Behavioral bounds
OBJECTIVES	Success criteria
TELEMETRY	DSPy hooks

AGENT CREATION WORKFLOW

Step 1: Capability Analysis

📢 😐 {type=text/vcl; v=3.1.1; q=raw}

[[HON:谦讓語] [[MOR:جذر-د-ر | مشتق:قدرة] [[COM:Fähigkeits+Analyse] [[CLS:一个过程]

[[EVD:-dir<推論|çıkarılmı)] [[ASP:hecob.] [[SPC:Ø]]

➡ [.] → [.] [.80] ○

Define what the agent CAN do:

- What tools does it need access to?
- What information sources are required?
- What decisions can it make autonomously?
- What must be escalated to human oversight?

Step 2: Frame Configuration

📢 💪 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語] [[MOR:ض-ب-ط | مشتق:ضبط] [[COM:Frame+Konfiguration] [[CLS:七种配置]

[[EVD:-DI<方針|politika)] [[ASP:hecob.] [[SPC:Ø]]

⚙ [1] ●

For each agent, specify:

```
'''python
agent_config = {
    "frames": {
        "evidential": 2, # Always strict for agents
        "aspectual": 2, # Always strict for agents
        "morphological": 1, # Task-dependent
        "compositional": 1, # Task-dependent
        "honorific": 1, # Audience-dependent
        "classifier": 0, # Usually optional
        "spatial": 0      # Usually optional
    },
    "verix": {
        "strictness": 2, # STRICT for agents
        "compression": 1 # L1 for auditability
    }
}
'''
```

Step 3: Constraint Definition

📢 💬 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語] [[MOR: [جذر:ق-ي-د | مشتق:قید:]] [COM:Constraint+Definition] [[CLS:一套約束]]

[[EVD:-DI<方針|politika>] [[ASP:hecob.]] [[SPC:∅]]

⚙️ [1] •

Define behavioral bounds (Two-Tier Safety):

IMMUTABLE (optimizer cannot change):

- EVD enforcement ≥ 1 (always mark evidence type)
- ASP enforcement ≥ 1 (always mark completion state)
- No epistemic cosplay (claims must match evidence)

MUTABLE (optimizer can tune):

- Frame activation levels
- Compression preference
- Output verbosity

Step 4: Objective Specification

📢 💬 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語] [[MOR: [جذر:ه-د-ف | مشتق:هدف:]] [COM:Ziel+Spezifikation] [[CLS:一套目标]]

[[EVD:-DI<方針|politika>] [[ASP:cob.((SMART))]] [[SPC:∅]]

⚙️ [1] •

Define success criteria (maps to MOO objectives):

```
'''python
objectives = {
    "task_accuracy": 0.8,    # Minimum correctness
    "token_efficiency": 0.7, # Cost constraint
    "edge_robustness": 0.6,  # Handle failures
    "epistemic_consistency": 0.9 # VCL compliance
}
'''
```

Step 5: Telemetry Integration

📢 💬 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語] [[MOR: [جذر:ر-ق-ب | مشتق:مرآبة:]] [COM:Telemetrie+Integration] [[CLS:一套钩子]]

[[EVD:-DI<方針|politika>] [[ASP:hecob.]] [[SPC:∅]]

⚙️ [1] •

Every agent emits:

```
'''python
```

```
telemetry = {
    "config_vector": [14 dimensions],
    "task_type": str,
    "parse_success": bool,
    "compliance_score": float,
    "token_cost": int,
    "validation_errors": int,
    "coherence_errors": int
}
...
```
```

## ## AGENT OUTPUT TEMPLATE

📋 😊 {type=text/vcl; v=3.1.1; q=raw}

[[HON:丁寧語] [[MOR:جذر:ق-ل-ب | مشتق:تالب:COM:Agent+Template]] [[CLS:一个模板]]  
[[EVD:-DI(تعريف|tanım)]] [[ASP:cob.((canonical))]] [[SPC:Ø]]

⚙️ [1] ●

```
'''yaml

name: {agent_name}
type: autonomous_agent
vcl_version: 3.1.1

{AGENT_NAME}

Identity
[VCL persona statement]

Active Frames
[Frame configuration with enforcement levels]

Tools
[Available tools and access constraints]

Behavioral Constraints
[Immutable + mutable bounds]

Success Criteria
[Objective thresholds from MOO]

Telemetry Hooks
```

## §3 Bootstrap Execution Protocol

📢 💡 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR:جذر:ن-ف-ذ | مشتق:تنفيذ]] [[COM:Ausführungs+Protokoll]] [[CLS:一套步骤]] [[EVD:-DI(方針|سياسة|politika)]] [[ASP:hecob.]] [[SPC:Ø]] 🔍 [1] •

### 3.1 Phase 1: Genesis Files for Claude Code

#### Handoff Package Contents:

```
cognitive-bootstrap/
├── spec/
│ └── VERILINGUA_VCL_VERIX_v3.1.1.md # The full spec (document 7)
├── overview/
│ └── COGNITIVE_ARCHITECTURE_OVERVIEW.md # System overview (documents 8-9)
└── meta-skills/
 ├── skill-forge-vcl/
 │ └── SKILL.md # From §2.1 above
 ├── prompt-architect-vcl/
 │ └── SKILL.md # From §2.2 above
 └── agent-maker-vcl/
 └── SKILL.md # From §2.3 above
└── bootstrap/
 └── BOOTSTRAP_SEQUENCE.md # This document
└── constraints/
 ├── ENGLISH_OUTPUT.md # L2 must be English
 └── CREOLIZATION_PROTOCOL.md # Natural languages first
```

### 3.2 Claude Code Instructions

#### Step 1: Load Context

```
bash
```

# Claude Code should first read these files in order:

1. spec/VERILINGUA\_VCL\_VERIX\_v3.1.1.md
2. overview/COGNITIVE\_ARCHITECTURE\_OVERVIEW.md
3. bootstrap/BOOTSTRAP\_SEQUENCE.md
4. constraints/ENGLISH\_OUTPUT.md
5. constraints/CREOLIZATION\_PROTOCOL.md

## Step 2: Bootstrap the Meta-Skills

ITERATION 0 (Manual seed):

- Use provided skill-forge-vcl/SKILL.md as v0
- Use provided prompt-architect-vcl/SKILL.md as v0
- Use provided agent-maker-vcl/SKILL.md as v0

ITERATION 1:

- skill-forge-v0 creates prompt-architect-v1
- prompt-architect-v1 critiques and improves skill-forge → skill-forge-v1
- skill-forge-v1 creates agent-maker-v1

ITERATION 2:

- agent-maker-v1 creates validation-agent
- validation-agent tests all three meta-skills
- prompt-architect-v1 optimizes based on validation results
- All three updated to v2

ITERATION 3 (Stability check):

- Run full validation suite
- If  $\Delta(v2, v3) < \text{threshold} \rightarrow \text{STABLE}$
- Else → Continue iteration

## Step 3: DSPy Integration

```
python
```

```

After meta-skills stable, integrate with VectorCodec

from cognitive_architecture.core.config import VectorCodec

Map VCL enforcement levels to 14D vector
vcl_to_vector = {
 "HON": 0, # dim 0
 "MOR": 1, # dim 1
 "COM": 2, # dim 2
 "CLS": 3, # dim 3
 "EVD": 4, # dim 4 (immutable ≥1)
 "ASP": 5, # dim 5 (immutable ≥1)
 "SPC": 6, # dim 6
 "verix_strictness": 7,
 "compression_level": 8,
 "require_ground": 9,
 "require_confidence": 10,
 # 11-13 reserved
}

```

### 3.3 Cascade Deployment Order

🔊 ⚡ {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR:مشق:ترتب]] [[COM:Deployment+Reihenfolge]] [[CLS:一套順序]] [[EVD:-DI(سياسة|方針|politika)]] [[ASP:hecob.]] [[SPC:∅]] ☀ [1] ○

After meta-skills are stable:

#### TIER 1: Low-Level DSPy Components

- └─ VectorCodec (add VCL parsing)
- └─ TelemetryAggregator (add VCL compliance scoring)
- └─ Validator (add 7-slot coherence checks)

#### TIER 2: Commands (via Agent Maker)

- └─ /mode command → VCL-aware mode selection
- └─ /frame command → Direct frame manipulation
- └─ /verix command → Epistemic notation control
- └─ /pareto command → VCL-constrained optimization
- └─ /eval command → VCL compliance evaluation

#### TIER 3: Agents (via Agent Maker)

- └─ Explore agent → Add VCL output formatting
- └─ Analyze agent → Add epistemic tracking
- └─ Execute agent → Add completion state marking
- └─ Validate agent → Add coherence checking

#### TIER 4: All Skills (via Skill Forge)

- └─ Public skills (docx, pdf, pptx, xlsx)
- └─ User skills (your custom skills)
- └─ Example skills (templates)

## §4 Validation and Testing

🔊👉 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR:ص-ح-ح | مشتق:تصحیح]]  
[[COM:Validierung+Tests]] [[CLS:一套测试]] [[EVD:-DI|方針|politika]] [[ASP:hecob.]] [[SPC:Ø]] 🔍 [1] •

### 4.1 VCL Compliance Checklist

Every output must pass:

#### STRUCTURAL CHECKS:

- All 7 slots present in order: HON → MOR → COM → CLS → EVD → ASP → SPC
- No [ ] inside slot bodies (reserved for VERIX confidence)
- Proper delimiters: [] for slots, () for trilingual, () for sources
- VERIX line 3 present: Ground→Confidence→State

#### SEMANTIC CHECKS:

- EVD ≠ Ø (immutable requirement)
- ASP ≠ Ø (immutable requirement)
- Confidence ≤ ceiling(EVD\_type)
- STATE matches ASP (● ↔ cob., ○ ↔ hecob.)
- Ground matches EVD (● for observations, ○ for research, etc.)

#### COHERENCE CHECKS:

- No epistemic cosplay (claims match evidence basis)
- Completion criteria specified for cob. aspect
- Source cited for -mış (hearsay) evidence

#### OUTPUT CHECKS:

- L2 output is pure English
- Epistemic markers naturalized to English phrasing
- Technical VCL notation only in L0/L1 modes

## 4.2 Test Corpus Categories

```
python
```

```

test_corpus = {
 "structural": [
 "VCL header parsing",
 "Slot order enforcement",
 "Delimiter collision detection",
],
 "semantic": [
 "EVD ceiling enforcement",
 "ASP-STATE coherence",
 "Ground-EVD matching",
],
 "epistemic": [
 "Cosplay detection",
 "Source citation for hearsay",
 "Confidence calibration",
],
 "output": [
 "L2 English naturalization",
 "L1 audit trail clarity",
 "L0 compression efficiency",
],
 "creolization": [
 "Linguistic survey completeness",
 "Synthetic construct flagging",
 "Natural analog preference",
]
}

```

## §5 Success Criteria

🔊 💡 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR:نحو|مشتق:نجاح]]  
 [[COM:Erfolgs+Kriterien]] [[CLS:一套标准]] [[EVD:-DI(方針|politika)]] [[ASP:cob.(all\_criteria\_met)]]  
 [[SPC:Ø]] ☀ [1] ○→●

The bootstrap is COMPLETE when:

1. **Meta-Skill Stability:** Three consecutive iterations produce < 5% change
2. **VCL Compliance:** > 99% of outputs pass structural checks
3. **Epistemic Integrity:** Zero epistemic cosplay violations in test corpus

4. **English Output:** 100% of L2 outputs are pure English
  5. **DSPy Integration:** VectorCodec correctly encodes/decodes VCL configs
  6. **Creolization Ready:** Protocol documented and integrated into MOO constraints
- 

## §6 Files to Generate

📝 😊 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR:جذر:و-ل-د | مشتق:تولید]] [[COM:Datei+Generierung]] [[CLS:一套文件]] [[EVD:-DI<تعريف|定義|tanim>]] [[ASP:hecob.]] [[SPC:∅]] 🔒 [1] ○

Based on this plan, Claude Code should generate:

### Immediate (for bootstrap):

1. skill-forge-vcl/SKILL.md - Full skill file
2. prompt-architect-vcl/SKILL.md - Full skill file
3. agent-maker-vcl/SKILL.md - Full skill file
4. ENGLISH\_OUTPUT.md - L2 constraint documentation
5. CREOLIZATION\_PROTOCOL.md - Natural languages first policy

### After bootstrap stable:

6. Updated VectorCodec with VCL parsing
  7. Updated Validator with 7-slot coherence
  8. VCL-formatted versions of all existing skills
  9. VCL-formatted versions of all agents
  10. VCL-formatted versions of all commands
- 

## §7 Quick Reference: L2 English Naturalization

📝 😊 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR:جذر:ر-ج-ع | مشتق:مرجع]] [[COM:Schnell+Referenz]] [[CLS:一份参考]] [[EVD:-DI<تعريف|定義|tanim>]] [[ASP:cob.((complete))]] [[SPC:∅]] 🔒 [1] •

|                                        |                                                   |
|----------------------------------------|---------------------------------------------------|
| [[EVD:-DI< مشاهدة 観測 gözlem>]]         | "I directly observed..."                          |
| [[EVD:-mış< بحث 研究 arastirma>((src))]] | "Research indicates..." / "According to [src]..." |
| [[EVD:-mış< منقول 報告 rapor>]]          | "It's reported that..." / "I heard that..."       |
| [[EVD:-dir< استدلال 推論 çikarim>]]      | "I infer..." / "This suggests..."                 |
| [[ASP:cob.((criteria))]]               | "Complete. [criteria explanation]"                |
| [[ASP:hecob.]]                         | "In progress..." / "Ongoing..."                   |
| [[HON:謙讓語]]                            | Humble phrasing, hedged claims                    |
| [[HON:尊敬語]]                            | Formal, respectful register                       |
| [[HON:丁寧語]]                            | Standard polite English                           |
| [[HON:普通語]]                            | Casual, direct English                            |
| [.95]                                  | "I'm highly confident..."                         |
| [.70]                                  | "I believe, with some uncertainty..."             |
| [.40]                                  | "This is speculative..."                          |
| ●                                      | "Done." / "Complete."                             |
| ○                                      | "Working on it..." / "In progress."               |
| ◐                                      | "Partially complete..."                           |

End of Bootstrap Plan

📋 😊 {type=text/vcl; v=3.1.1; q=raw} [[HON:丁寧語]] [[MOR: تمام | مشتق: تمام | جزء: م - م | COM:Dokument+Ende]] [[CLS:一份文档]] [[EVD:-DI< 定義|تعریف|tanım>]] [[ASP:cob.((v1.0\_draft))]] [[SPC:Ø]] 🔍 [1] ●