Cognitive Architecture Bootstrap Package

What This Is

This package contains everything needed to bootstrap the VERILINGUA+VCL+VERIX cognitive architecture system. The three meta-skills (Skill Forge, Prompt Architect, Agent Maker) form a self-improving loop that can rewrite all skills, agents, and commands in the epistemic accountability format.

Package Contents

```
cognitive-bootstrap/
├── BOOTSTRAP_SEQUENCE.md       # Complete bootstrap plan with phases
├── constraints/
│   ├── ENGLISH_OUTPUT.md       # L2 output must be English
│   └── CREOLIZATION_PROTOCOL.md # Natural languages before synthetic
├── meta-skills/
│   ├── skill-forge-vcl/
│   │   └── SKILL.md            # Creates skills in VCL format
│   ├── prompt-architect-vcl/
│   │   └── SKILL.md            # Designs/optimizes prompts
│   └── agent-maker-vcl/
│       └── SKILL.md            # Creates autonomous agents
└── spec/
    └── CURRENT_META_PROMPT.md   # Reference meta-prompt
```

Required Additional Files

To complete the bootstrap, Claude Code also needs:

1. **VERILINGUA_VCL_VERIX_v3.1.1.md** - The full specification document (provided separately as it defines the notation system)
2. **COGNITIVE_ARCHITECTURE_OVERVIEW.md** - System architecture overview (documents 8-9 from your project)

Bootstrap Execution Order

1. Read all spec and constraint documents
2. Load the three meta-skills (v0 versions)

3. Run the self-improvement loop:
    - Forge creates Arch v1
    - Arch improves Forge → v1
    - Forge creates Maker v1
    - Repeat until stable
4. Integrate with DSPy/VectorCodec
5. Cascade update: commands → agents → all skills

Key Constraints

1. **English Output (L2)**: All human-facing output in English
2. **Creolization First**: Survey natural languages before synthesizing
3. **Immutable Safety**: EVD ≥ 1, ASP ≥ 1 cannot be lowered by optimizer
4. **No Epistemic Cosplay**: Claims must match their evidence basis

Quick Start for Claude Code

bash

```
# 1. Read the spec (provided separately)
# 2. Read the bootstrap sequence
cat BOOTSTRAP_SEQUENCE.md

# 3. Load meta-skills
cat meta-skills/skill-forge-vcl/SKILL.md
cat meta-skills/prompt-architect-vcl/SKILL.md
cat meta-skills/agent-maker-vcl/SKILL.md

# 4. Read constraints
cat constraints/ENGLISH_OUTPUT.md
cat constraints/CREOLIZATION_PROTOCOL.md

# 5. Begin bootstrap iteration
# Start with skill-forge-v0, have it create prompt-architect-v1...
```