

## Task:

# Soham Kotkar — Sovereign LM Bridge + Multilingual KSML Core (MCP + RL + Vaani-ready)

Duration: Oct 28 – Nov 2

Goal: Build the sovereign multilingual reasoning bridge that connects Bhavesh's LM Core, Vaani TTS, and Gurukul/Uniguru front-end — fully KSML aligned, RL-updatable, and MCP-streaming ready.

## One-line Objective

Create a live multilingual reasoning core that listens to Bhavesh's LM responses, refines them via RL-based language alignment, and streams KSML-tagged results + speech-ready text to Karthikeya's Vaani system.

## Core Deliverables

### KSML Semantic Alignment Engine

- Implement /align.ksml service (FastAPI).
- Accepts raw LM text (from Bhavesh's system) and adds:

```
{  
  "intent": "...",  
  "source_lang": "hi",  
  "target_lang": "en",  
  "karma_state": "sattva/rajas/tamas",  
  "semantic_roots": ["dhātu", "artha", "bhava"]  
}
```

- Lightweight Sanskrit-root tagging via predefined lookup JSON (ksml\_roots.json).

### MCP-Driven Feedback Stream

- Integrate with MCP connectors to pull live examples (user prompts + corrections).
- Auto-store into /data/feedback\_stream.jsonl.
- Every feedback cycle updates a small in-memory policy (Q-table or bandit style).

### RL Self-Improvement Loop

- Add /rl.feedback endpoint:  
Accepts { prompt, output, reward } → updates local adapter delta or policy table.
- Run periodic reward-based adjustments (no full retraining).
- Sync logs to s3://bhiv/rl\_feedback/sovereign\_core/.

## Vaani Compatibility Layer

- Create /compose.speech\_ready endpoint:  
Converts aligned text → prosody-optimized JSON for Karthikeya's TTS engine.

```
{
  "text": "The answer is...",
  "tone": "calm",
  "lang": "en",
  "prosody_hint": "gentle_low"
}
```

- Confirm with Karthikeya that tone + prosody hint fields map correctly.

## Multilingual Reasoning Bridge

- Add connector to Bhavesh's /compose.final\_text API.
- Automatically run alignment + feedback + prosody preparation in one flow.
- Expose /bridge.reason endpoint → gives unified output (text + KSML + prosody).

## System Integration + Logging

- Store everything under `/logs/ksml_bridge.jsonl` with timestamps, source trace\_id.
- Maintain latency under 2s (end-to-end pipeline).
- Use <4GB VRAM, run smoothly on RTX 4050.

## File & Folder Plan

```
sovereign_core/
├── api.py                    # FastAPI endpoints
(align.ksml, rl.feedback, compose.speech_ready)
├── ksml/
│   └── aligner.py          # intent + karma + root
tagging
```

```

|   |   | ksm1_roots.json      # Sanskrit roots + meanings
|   |   | rl/
|   |   | | policy.py         # simple RL/bandit for reward
learning
|   |   | | feedback_logger.py # logs reward updates
|   |   | | bridge/
|   |   | | | bhavesh_connector.py # connects to /
compose.final_text
|   |   | | vaani_adapter.py  # maps tone/prosody for
speech-ready output
|   |   | | mcp/
|   |   | | | stream_client.py # fetch feedback samples
|   |   | | | config.yml
|   |   | | logs/
|   |   | | | ksm1_bridge.jsonl
|   |   | requirements.txt
|   |   | README.md

```

## Coordination

Area	Collaborator	Responsibility
LM Response Source	Bhavesh	Provide API endpoint and example outputs
TTS Mapping	Karthikeya	Confirm tone & prosody schema
Feedback & RL Storage	Vijay	S3/NAS endpoint for reward uploads
MCP Streams	Nipun	Core dataset connectors
Testing & Task Bank	Vinayak	Basic pipeline validation

## Timeline (5 Days)

Day	Focus
Day 1	Setup repo, connect to Bhavesh API, stub endpoints
Day 1	KSML aligner + MCP stream
Day 2-3	RL feedback + reward logging
Day 2-3	Vaani compatibility + speech-ready adapter

Day 3-4	Unified reasoning bridge /bridge.reason
Day 4-5	Smoke tests, optimization (<2s latency), full documentation

## Acceptance Criteria

Metric	Target
Latency (end-to-end)	$\leq 2s$
GPU Memory (RTX 4050)	$\leq 4\text{ GB}$
Languages Supported	$\geq 10$ live MCP-streamed
KSML Tag Accuracy	$\geq 85\%$ consistency
RL Reward Updates	Visible in logs
Prosody + Speech Output	Functional with Vaani
Documentation	Clear + reproducible

## After Completion

This task completes Layer 2 of the Gurukul Sovereign LM Stack —

Layer 1: Bhavesh's LM Core

Layer 2: Soham's Multilingual Reasoning Bridge

Layer 3 (Next): Karthikeya's Vaani Expressive RL-TTS

All three connect under the BHIV Central Cognitive Mesh (managed by Vinayak).