Opic Language Documentation

Opic is a self-hosting language for distributed computation with cryptographic trust.

Features

- **Self-hosting**: Opic defines itself in `.ops` files
- **Cryptographic trust**: Each voice is signed with a certificate
- **Field equations**: Programs evolve under field dynamics
- **Compositional**: Voices compose naturally into chains

> **Note:** Voices are opic's fundamental unit of computation. They transform inputs to outputs.

OPS Syntax Example

Here's a simple opic voice with syntax highlighting:

```
voice greet / {name -> "Hello " + name -> greeting} voice main / {greet "world" -> greet}
```

This demonstrates opic's compositional elegance: voices transform inputs to outputs, and compose naturally.

Field Equations

Opic's architecture expresses field dynamics through mathematical equations:

Field Equation Exchange: Learning Energy Units computed from time, coherence, and validation

$$f(\Delta t, \Delta\Phi, \text{proof\_of\_care}) = \sum_i w_i * (t_i + \Phi_i + v_i)$$

Where: tau = (t, Phi, v)

> **Tip:** The field equation computes Learning Energy Units from time, coherence, and validation.

The coherence equation describes how fields evolve:

Field coherence evolution: rate of change equals divergence of current plus sources

$$\frac{d\Phi}{dt} = \text{div } J + S$$

Where: Phi = field state, J = current, S = source

Cryptographic Seals

Each opic voice can be cryptographically signed:

CERTIFICATE

Realm: **opic_realm**

CA: **opic_ca**

Signature: sha256:abc123def456...

WITNESS

Realm: **opic_realm**

CA: **opic_ca**

Signature: sha256:witness789...

Warning: Always verify signatures before executing voices from untrusted sources.

Architecture

Opic's architecture naturally expresses the duality at the heart of analytic number theory:

- **Left Flank — Category (Discrete)**: voices compose into a spectrum of prime morphisms
- **Right Flank — Field (Continuous)**: coherence evolves under field equations
- **Bridge — Certificate Operator**: a unitary transformation equating the two halves

Note: This duality mirrors the Riemann zeta function's functional equation.

The zeta equation demonstrates this symmetry:

Riemann zeta functional equation: symmetry across the critical line

$$\zeta(s) = \chi(s) * \zeta(1 - s)$$

Where: s = 1/2 + i*t

Conclusion

Opic combines mathematical elegance with practical cryptographic trust, enabling distributed computation with built-in verification.

SIGNED

Realm: **opic_realm**

CA: **opic_ca**

Signature: sha256:final...