CALM

COMMON ARCHITECTURE
LANGUAGE MODEL

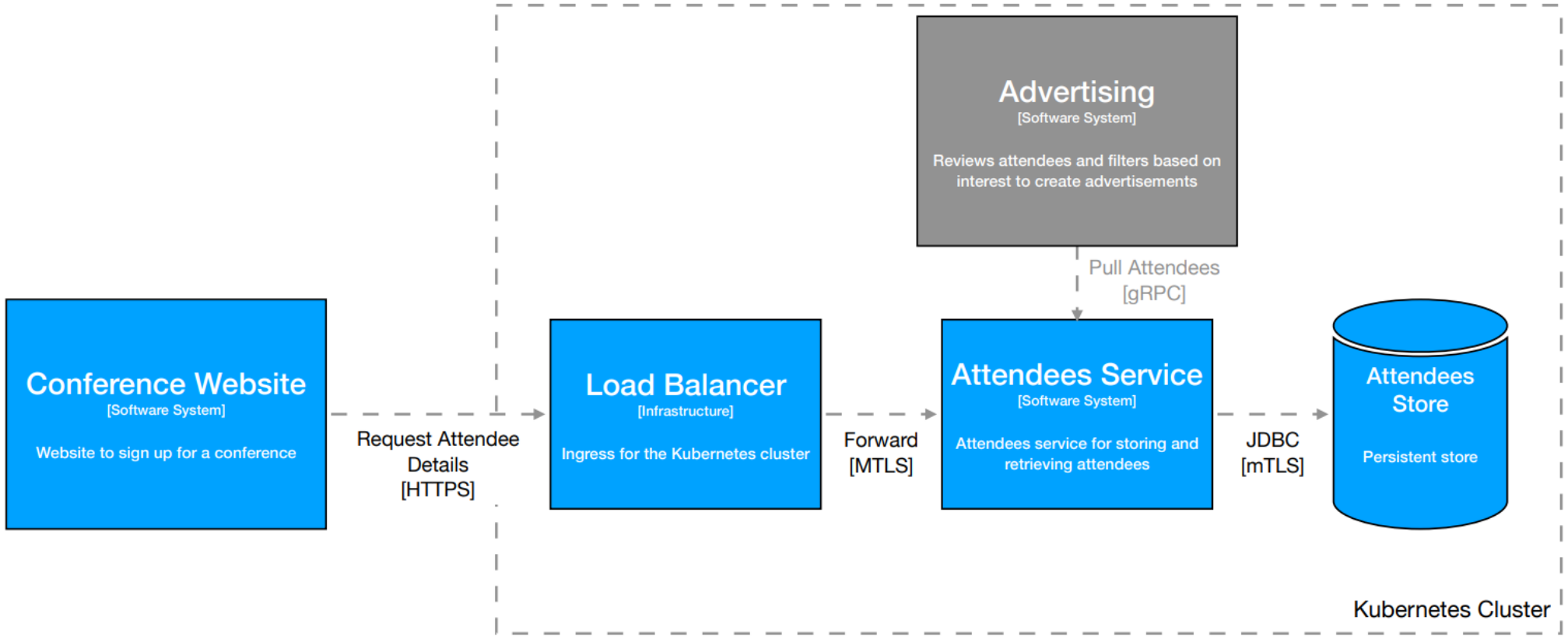
Why CALM?

- Architecture lives outside the developer workflow
- Captured on whiteboards, in Visio diagrams and PowerPoints
- Unlike requirements, code, tests, or deployments, architecture is not systematically measurable
- This leads to:
  - Uncertainty about whether what was designed is what was built.
  - Inability to detect drift over time.
  - Missed opportunities for automation and control.

What is CALM?

- A JSON Meta Schema to describe system architectures
  - Nodes >> Solid Boxes
  - Relationships >> Arrows + Dashed Boxes
  - Interfaces >> Features exposed by a node
  - Controls >> Non-functional requirements
  - Flows >> Movement through the logical architecture
  - Metadata >> Any additional information
  - Decorators >> Cross cutting concerns
  - Timelines >> Architecture changes through time

CALM – Sample Architecture

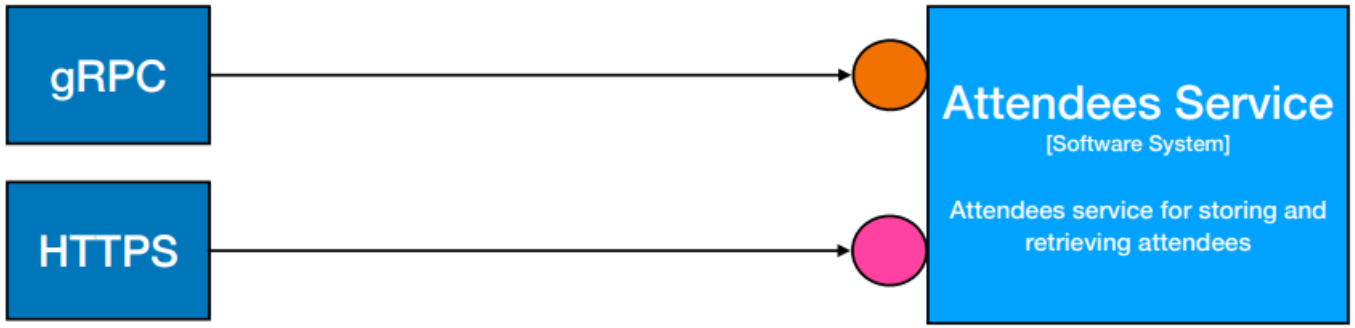


CALM - Nodes


Node
Conference Website
[Software System]
Website to sign up for a conference
{
"unique-id": "conference-website",
"node-type": "system",
"name": "Conference Website",
"description": "Website to sign up for
conference",
},
JDBC
[mTLS]
Node
Attendees
Store
Persistent store
Kubernetes Cluster
Node

CALM - Relationships

CALM - Interfaces




```
{
    "unique-id": "attendees-store",
    "name": "Attendees Store",
    "description": "Persistent storage for attendees",
    "node-type": "database",
    "interfaces": [
      {
        "unique-id": "database-image",
        "image": "[[ IMAGE ]]"
      },
      {
        "unique-id": "database-port",
        "port": -1
      }
    ]
  },        …
```

CALM - Controls

```
{
  "$schema": "http://json-schema.org/draft/2020-12/schema",
  "title": "NIST Document Metadata",
  "type": "object",
  "properties": {
    "documentNumber": {
      "type": "string",
      "description": "The official NIST document number, e.g., 'NIST SP 800-207'"
    },
    "title": {
      "type": "string",
      "description": "The full title of the NIST document"
    },
    "status": {
      "type": "string",
      "enum": ["Final", "Draft", "Superseded", "Withdrawn"],
      "description": "The current status of the document"
    },
    …
```

CALM - Flows


```
"flows": [
   {
    "unique-id": "conference-registration-flow",
    "name": "Conference Registration Flow",
    "description": "Complete user journey for registering for the conference, from initial web access through data
persistence",
    "transitions": [
     {
      "relationship-unique-id": "conference-website-load-balancer",
      "sequence-number": 1,
      "summary": "User submits conference registration form via the website",
      "direction": "source-to-destination"
     },
     {
      "relationship-unique-id": "load-balancer-attendees",
      "sequence-number": 2,
      "summary": "Load balancer forwards registration request to attendees service",
      …
```

CALM - Decorator

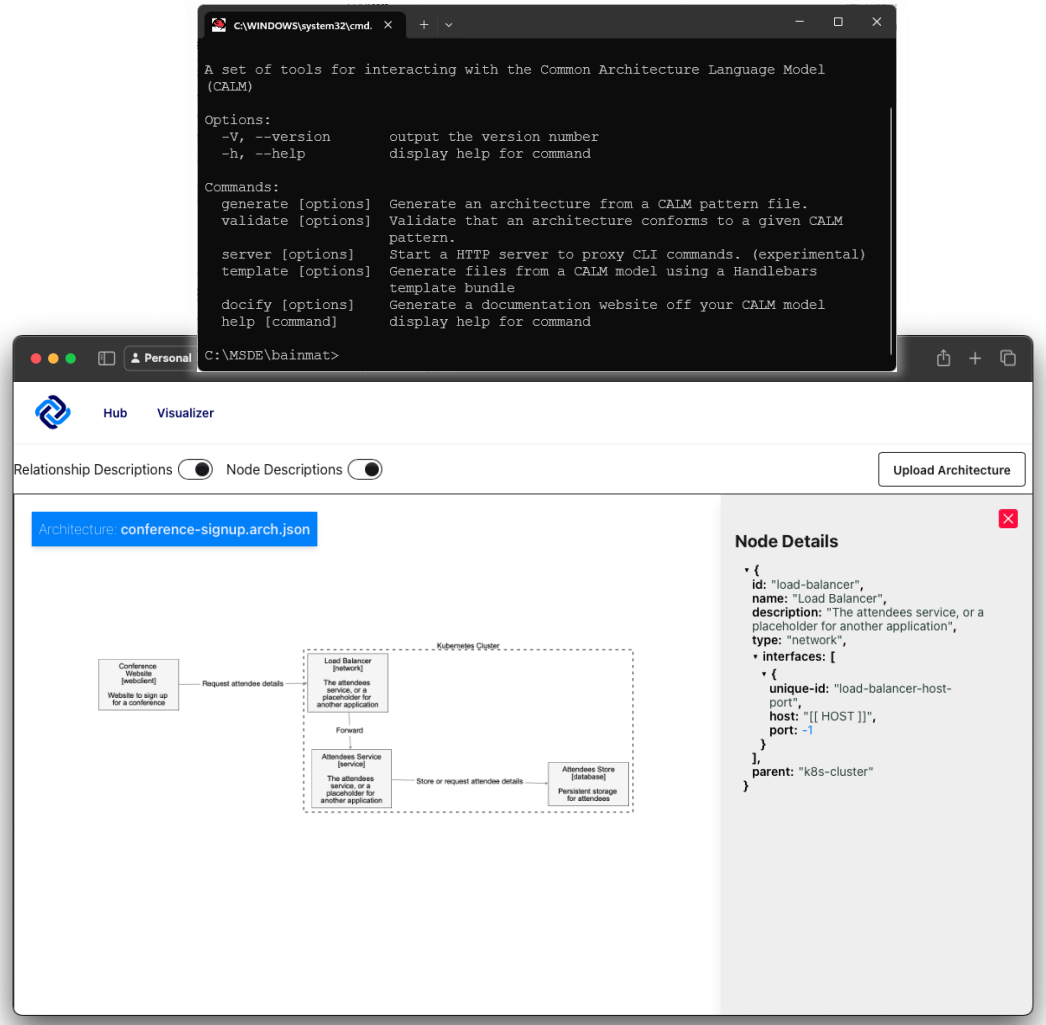
```
{
   "$id": "https://calm.finos.org/release/1.2/meta/kubernetes.decorator.schema.json",
   "title": "CALM Kubernetes Deployment Decorator Schema",
   "type": "object",
   "properties": {
       "data": {
       "type": "object",
       "properties": {
           "kubernetes": {
              "type": "object",
              "properties": {
                  "helm-chart": { "type": "string", "pattern": "^[a-z0-9-]+:[0-9]+\\.[0-9]+\\.[0-9]+$“ },
                   "cluster": { "type": "string" },
                   "namespace": { "type": "string" }
              },
              "required": ["helm-chart", "cluster"],
              "additionalProperties": false
       }
   },
   "required": ["kubernetes"]
}

       …
```

CALM - Timeline


```
{
 "current-moment": "v2-separated-db",
 "moments": [ {
    "unique-id": "v1-monolith",
    "node-type": "moment",
    "name": "Initial Monolithic Architecture",
    "description": "The original monolithic application serving all functionality",
    "valid-from": "2020-01-15",
    "details": { "detailed-architecture": "architectures/v1-monolith.json“ },
    "adrs": [ https://github.com/org/repo/blob/main/docs/adr/001-initial-architecture.md ]
  }, {
    "unique-id": "v2-separated-db",
    "node-type": "moment",
    "name": "Database Separation",
    "description": "Separated the database into dedicated instances for improved scalability",
    "valid-from": "2022-06-20",
    "details": { "detailed-architecture": "architectures/v2-separated-db.json“   },
    "adrs": [ https://github.com/org/repo/blob/main/docs/adr/005-database-separation.md ]
  } ],
 "metadata": {
  "title": "Payment System Architecture Timeline",
  "description": "Evolution of the payment processing system from monolith to event-driven microservices",
  "owner": "Platform Architecture Team",
 }
}
```

CALM – Patterns + Tools

- Patterns
  - Reusable, pre-approved, opinionated architectures
  - Drive standardisation + automation
  - The **superpower** of CALM
- A set of tools for working with CALM
  - CLI (published to NPM) >> generate architectures from patterns, validate architectures, create human and machine readable documents
  - CALM Hub >> NPM for CALM Artefacts, Visualisation
  - CALM Copilot Chat Mode >> Make Copilot your Architecture SME



Where are we?

- CALM JSON Meta Schema & CLI published v1.0 in August 2025
  - v1.1 introduced in November 2025 and v1.2 in February 2026
  - Install via NPM (`npm install -g @finos/calm-cli`) or Homebrew (`brew install calm-cli`)
- VS Code Extension published in October 2025
  - https://marketplace.visualstudio.com/items?itemName=FINOS.calm-vscode-plugin

AI Support

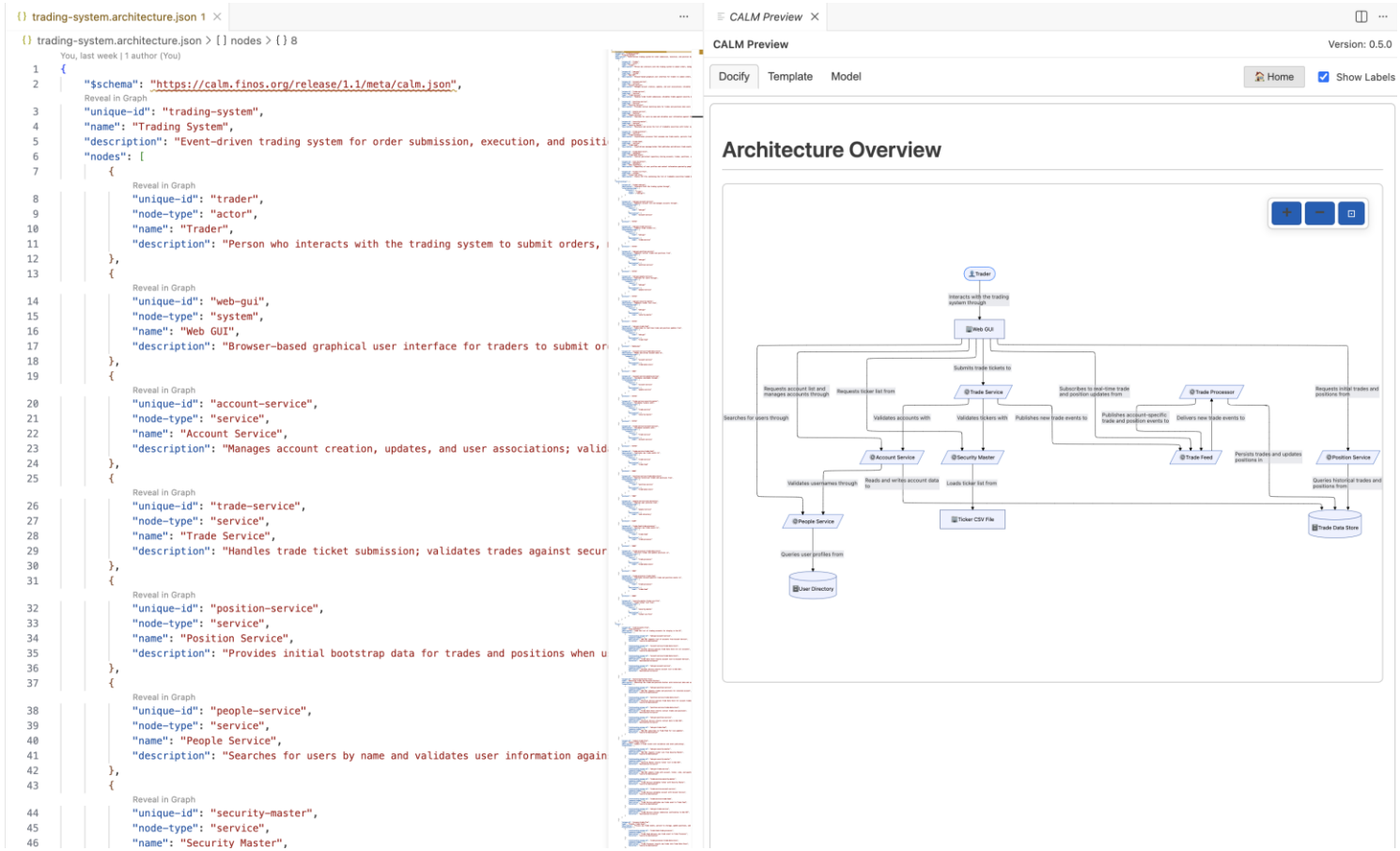
- Turn your CALM Model into a peer architect with CALM AI Support

LLM Supported	IDE
LLM available in Github Copilot	VSCode
Claude family of LLM	KIRO
Claude family of LLM	Claude Code - CLI

- Allows you to create all CALM components without typing any JSON
- Turns your model into an interactive query tool
- See https://calm.finos.org/working-with-calm/calm-ai-tools for more details

AI Support (cont'd)

- Use natural language to create your nodes
  - Create nodes representing the following services
- Instruct the LLM on how they should be connected
  - Create a connects relationship between node A and B, using B's interface
  - Create a deployed-on relationship between node A and container B
- Ask the LLM questions
  - Identify single points of failure?
  - What happens if Node B goes down?



See https://calm.finos.org/tutorials/build-a-calm-architecture/define-initial-architecture for details

Where are we going?

- Adding more capabilities
  - Deployment & Source Control lenses
- Further use-cases:
  - Policy as Code
  - Architecture drift
  - Partner with Moderne to further developer Prethink - https://docs.moderne.io/user-documentation/moderne-platform/getting-started/prethink/

Like what you see?

- Get involved, we're looking for early adopters who are keen to learn CALM and provide feedback or open source contribution
  - Good First Issues: https://github.com/finos/architecture-as-code/issues?q=state%3Aopen%20label%3A%22good%20first%20issue%22
- CALM tutorials: https://calm.finos.org/tutorials/
- Two regular catch ups:
  - **Monthly Architecture as Code Working Group Meeting:** fourth Tuesday of every month @ 11:00 AM to noon Eastern
  - **Weekly Architecture as Code Office Hours:** Thursday @ 10:30AM to 11:30AM
- Head over to https://calendar.finos.org/ to sign up

Who Am I?

- Khalid Elsawaf – Executive Director at Morgan Stanley
- Lead Architecture for Secured Financing Technology Post Trade
- Montreal Lead for Prime Brokerage & Secured Financing Technology
- Email: khalid.elsawaf@morganstanley.com
- LinkedIn: https://ca.linkedin.com/in/khalid-elsawaf-33560811