Building Streaming Data Applications With Properly-Streaming Architectures

Rohit Bose
rohit@swim.inc
brohitbrose (GitHub, Twitter, Reddit)

Agenda

- Discuss streaming data applications
- Showcase an app built with a truly streaming architecture
- Technical not-so-deep dive
- Q/A

Streaming Data Application Requirements

- **Stateful microservices** that perform *arbitrary logic* with *memory-latency access* to required contexts
- **Streaming APIs**—react instantly to changes; don't waste cycles awaiting them
- **UIs** and downstream services inherently **real-time**

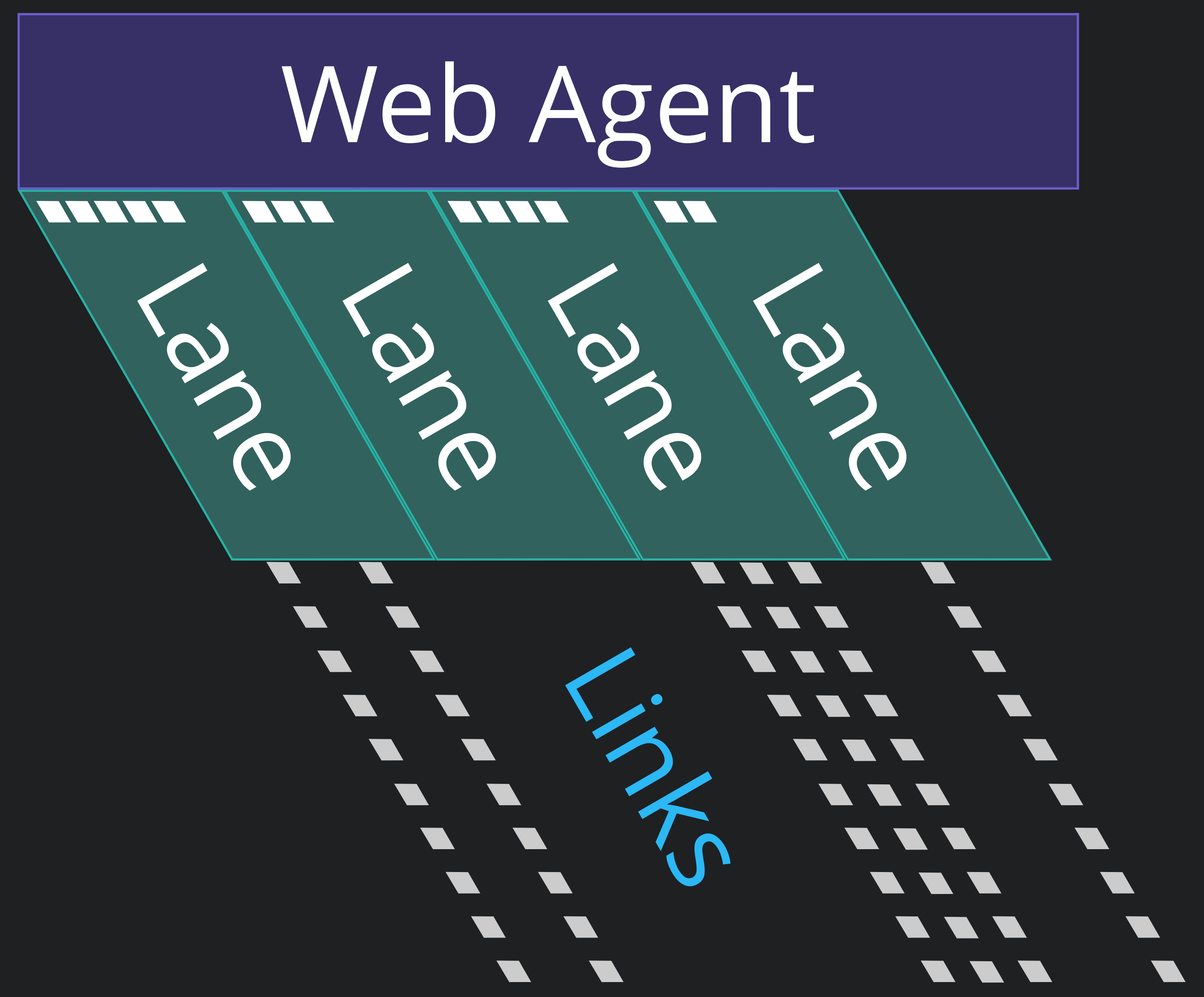
Demands From Modern Apps

- **Continuousness**: streaming with minimal latency
- **Autonomy**: actionability of **stateful microservices** is unaffected by higher-level algorithmic restrictions
- **Observability**: coordination-free inspectability, possibly including internal state
- Not mutually exclusive—*loosening* these properties *conflicts with the needs of* modern streaming apps

One Solution: Web Agents

- The **Web Agent architecture**: Swim's general-purpose *distributed object model* of **stateful microservices** for building end-to-end streaming applications
- First-class citizens of the World Wide Web that strictly communicate over **streaming APIs** (WARP, not REST)
- **Real-time UIs** and other actionable actors *designed around* streams

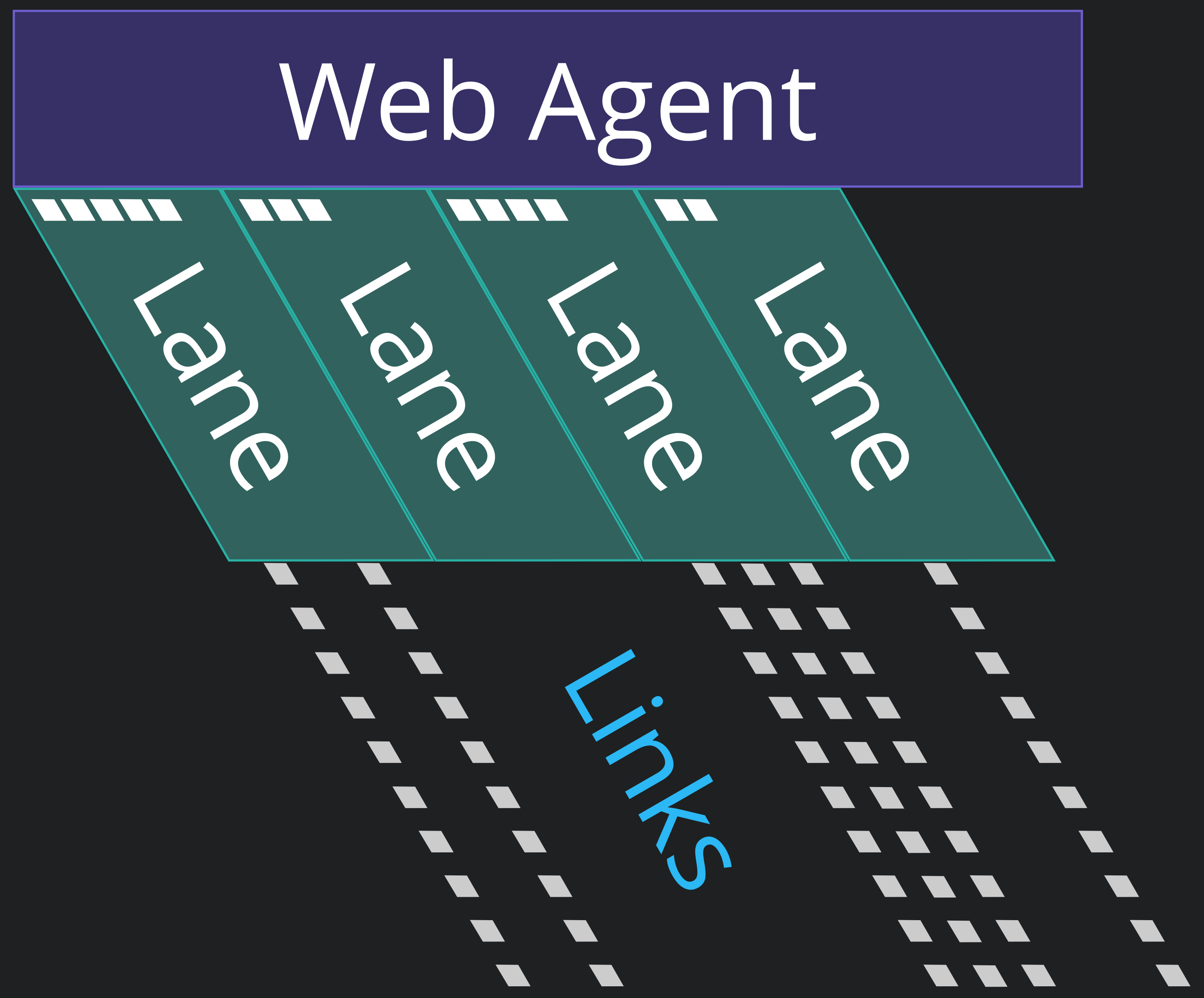
Web Agent Model At A Glance



- *Web Agents*: distributed, deployment-agnostic, observable, composable stateful “objects”
- *Lanes*: “members” with actionable lifecycle callbacks
- *Links*: “references” with actionable lifecycle callbacks
- Express apps as logical objects, and *reactions to changes to* those objects

swim

Web Agent Analogies

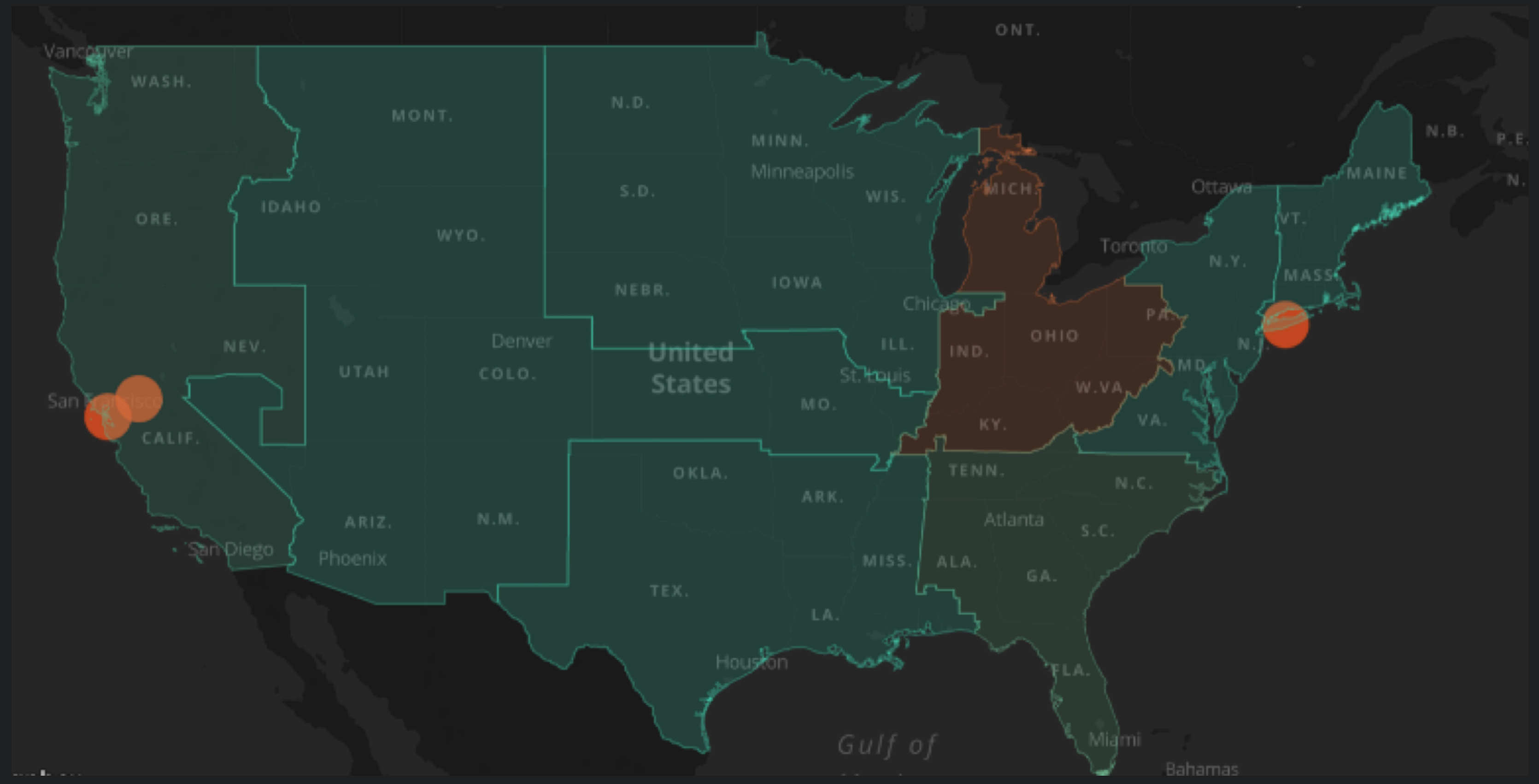


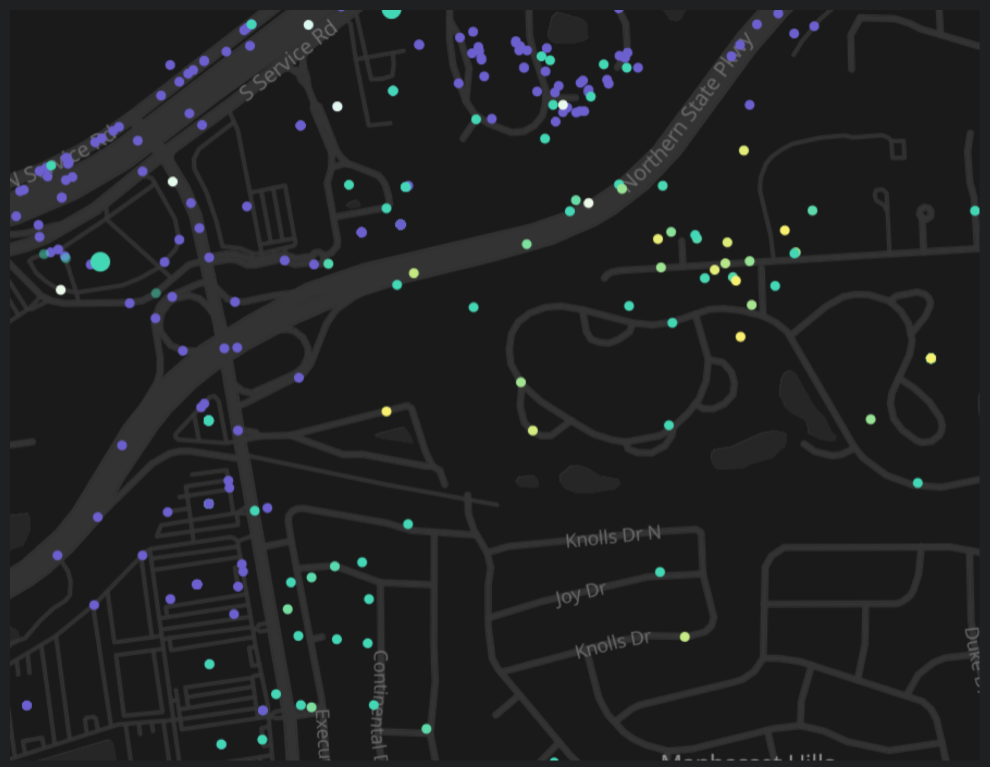
	Agent	Lane	Link
OOP	Object	Member	Reference
REST	Endpoint	Method	Request
Database	Row	Column	Relation
Message Broker	Namespace	Topic	Subscription
Actor Model	Actor	Mailbox	Messages
Operating System	Process	File	File Handle

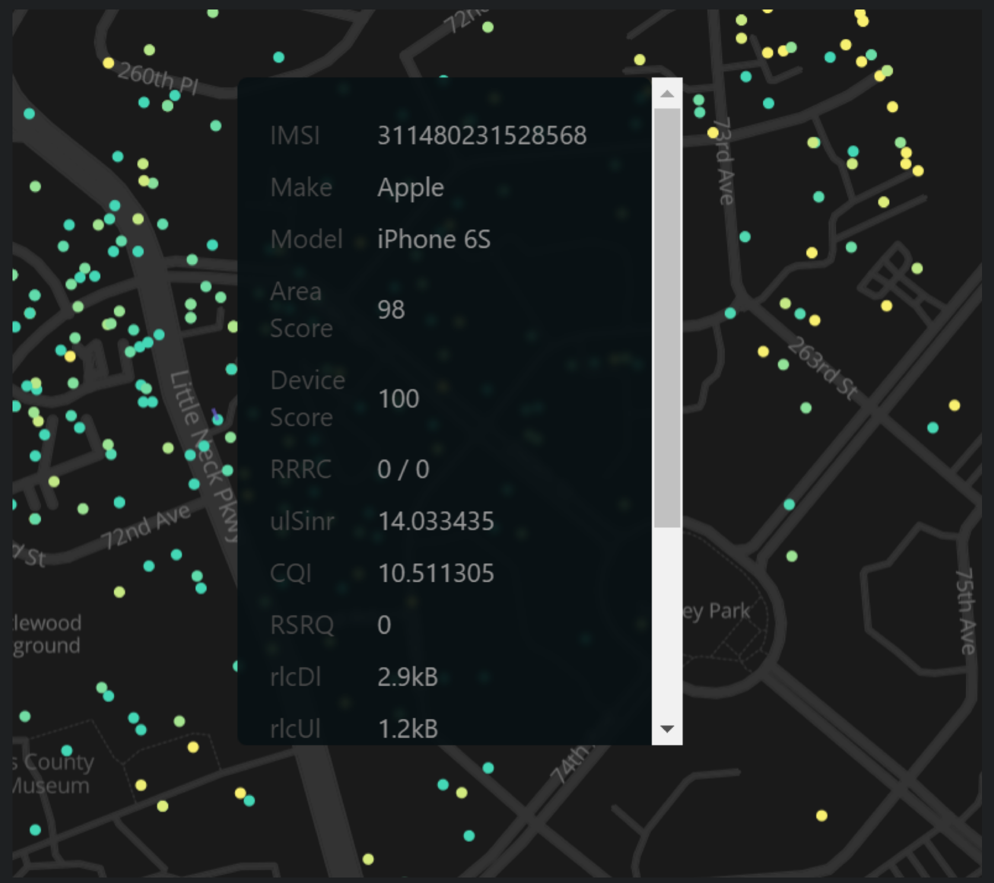
swim

Demo Application Spec

- **Continuous** monitoring and analytics of cell towers (~70k) across the nation
- Direct **observability** of, and computations upon, arbitrary subsets and aggregations
- Time-critical error identification
- Real-time visualization
- **Autonomous** actionability
- Accurate internal metrics reporting

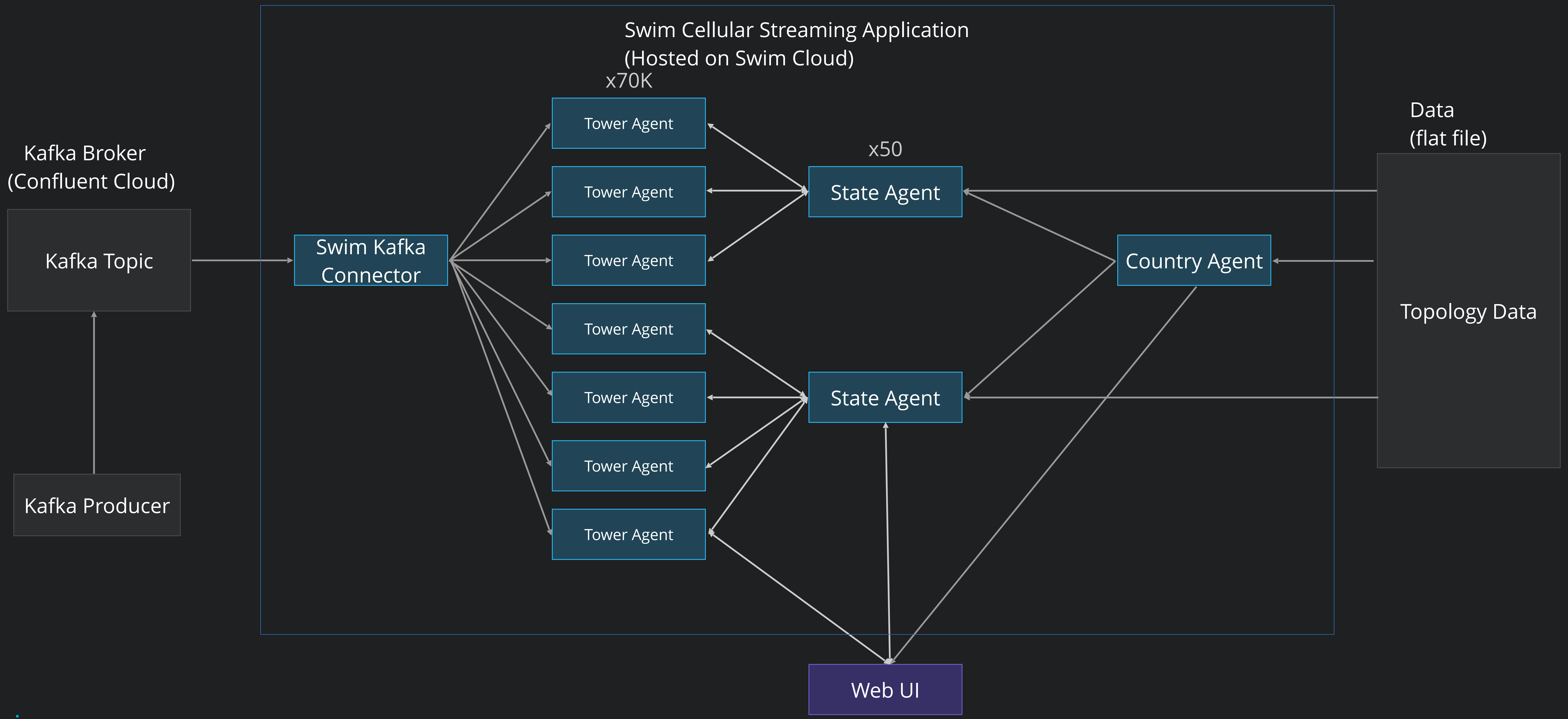






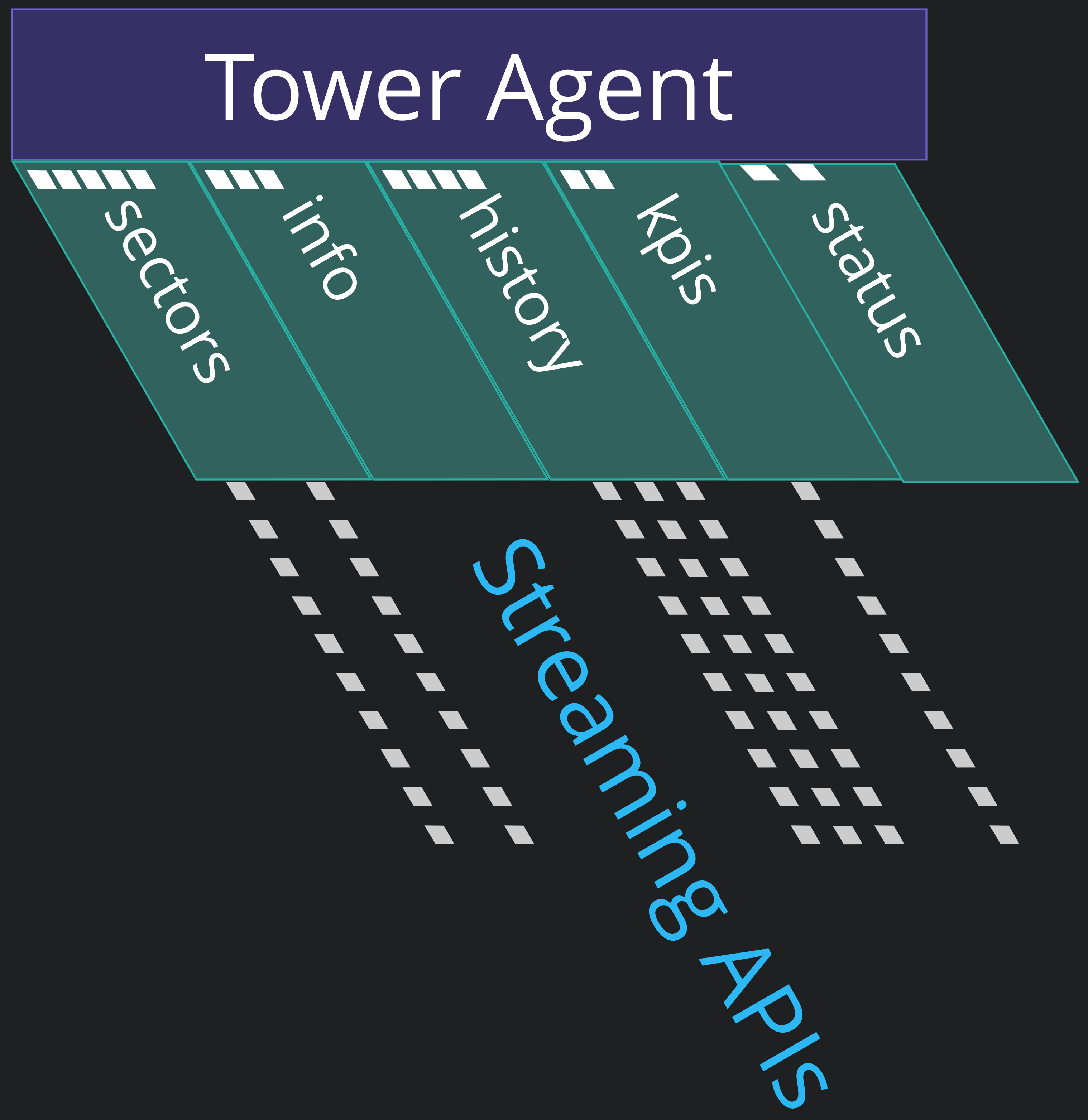
IMSI	311480231528568
Make	Apple
Model	iPhone 6S
Area Score	98
Device Score	100
RRRC	0 / 0
ulSinr	14.033435
CQI	10.511305
RSRQ	0
rlcDl	2.9kB
rlcUl	1.2kB

Cellular Application Architecture



swim

Tower Agent

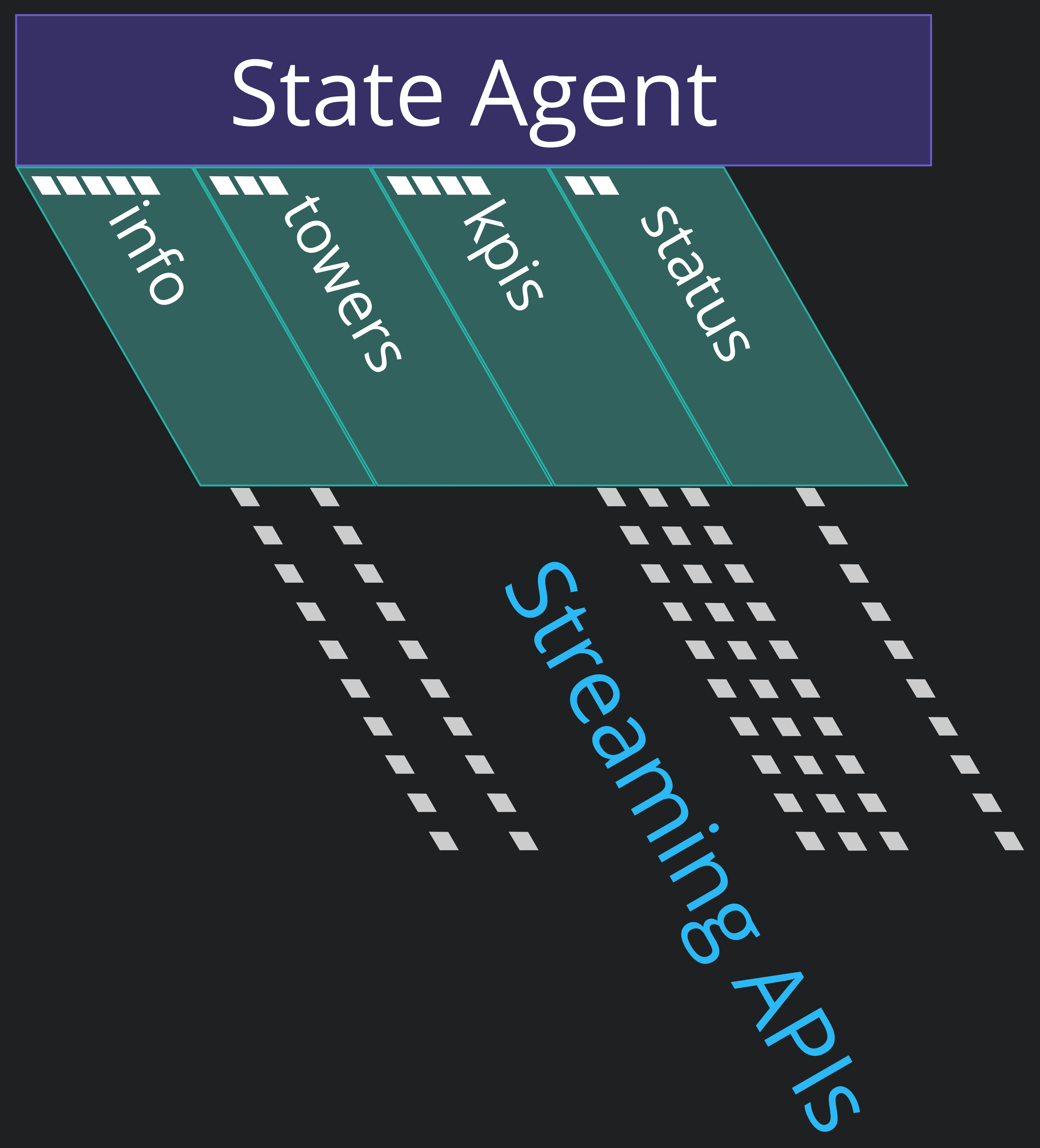


- Receives network metrics for a given tower from the Swim Kafka Connector

- Statefully models the current state of the tower

- Continuously analyzes and computes KPIs and status as it changes in real-time

State Agent



- Receives the real-time condition of every tower within a U.S. state (by *linking* to Tower Agents)

- Statefully models the current condition of its corresponding U.S. state

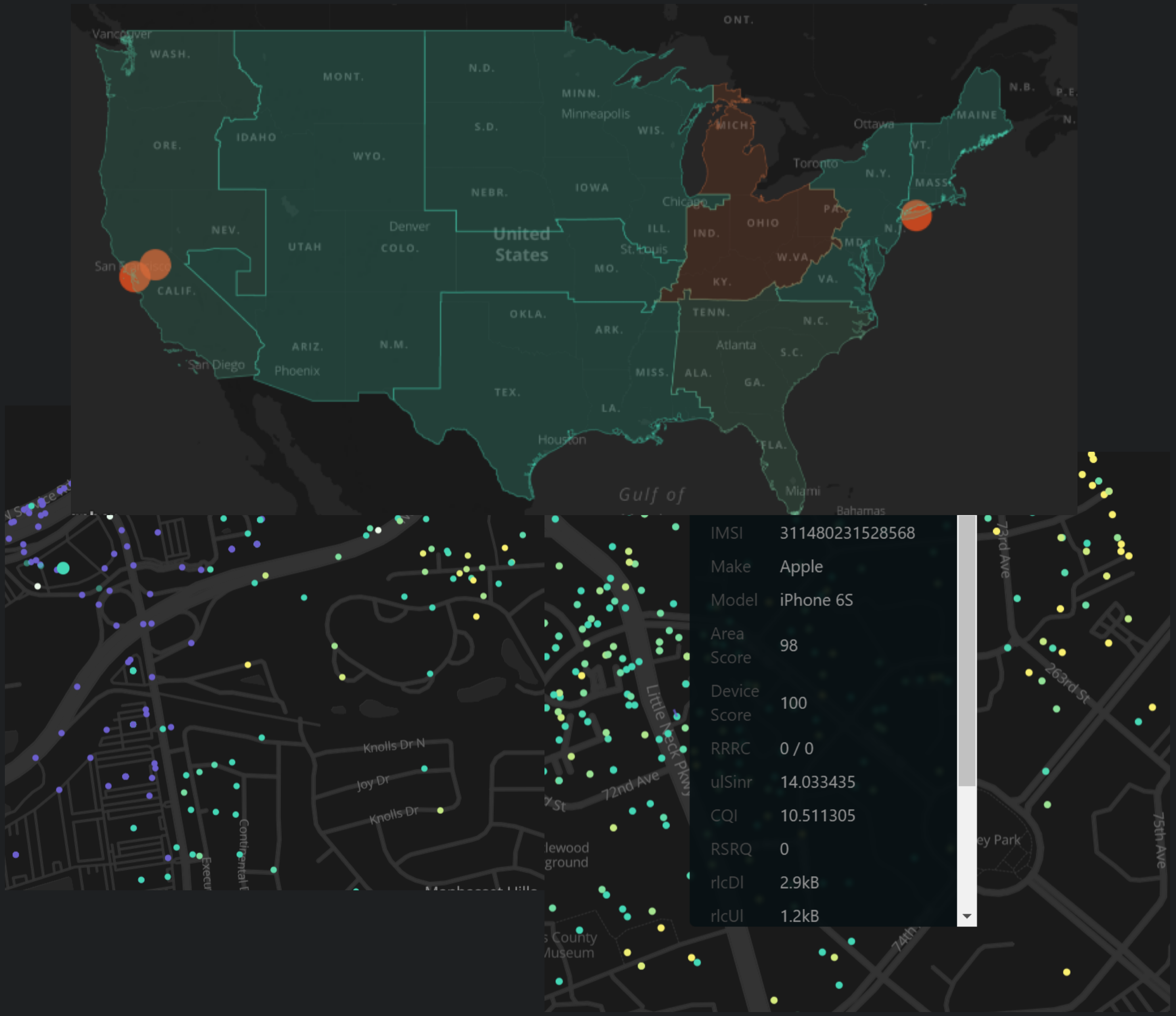
- Continuously analyzes and computes aggregate KPIs and statuses as it changes in real-time

Continuum UI

- Communicates–in real-time–directly with Web Agents using the same WARP protocol that Web Agents use to communicate with each other

- The UI *is* just another Web Agent (that happens to be a User Agent)

- The UI dynamically links to just the Web Agents that are relevant to the current view



IMSI	311480231528568
Make	Apple
Model	iPhone 6S
Area Score	98
Device Score	100
RRRC	0 / 0
ulSinr	14.033435
CQI	10.511305
RSRQ	0
rlcDl	2.9kB
rlcUl	1.2kB

Today's Key Takeaways

- Proper data streaming applications consist of **stateful microservices**, **streaming APIs**, and **real-time downstream UIs / processes**
- Challenges facing data streaming applications are **continuousness**, **autonomy**, and **observability**
- Today's demo that uses **Web Agents** to achieve these goals available at: https://continuum.swim.inc/cellular-confluent/#atlas
- Other useful links
  - https://swim.inc
  - https://www.swimos.org/
  - https://github.com/swimos/
  - https://traffic.swim.inc/
  - http://transit.swim.inc/

swim