REALITY, PERCEPTION AND NIHILISM

AS FAR AS THE LAWS OF MATHEMATICS REFER TO REALITY, THEY ARE NOT CERTAIN, AND AS FAR AS THEY ARE CERTAIN, THEY DO NOT REFER TO REALITY

WHAT REALLY IS REALITY?

IS ANYTHING REAL AT ALL?

- We, the humans, construct our reality based on external stimuli
- We react to each input and apply it to our model of reality
- Some of these reactions are externally visible for others to observe
- Others use these visible reactions as their own inputs
- They update their own model of reality by reacting to these inputs
- At no time do we reach into other people's brains to update reality
- Reality therefore is just an aggregation of events in time domain
- $R_t = \{ \text{Event} \mid \text{Time}_{\text{Event}} <= t \}$

WHAT IS PERCEPTION?

NOT THE MATRIX BUT PRETTY CLOSE

Perception can lag behind reality (Latency)

It can sometimes even conflict with reality (Inconsistency)

Occasionally it may have incomplete data (Fidelity)

Perception is (usually an acceptable) model of reality:

$$M(R_t) \approx \{ \text{Event} \mid \text{Time}_{\text{Event}} <= t \}$$

QUESTIONING REALITY

THE LAMBDA ARCHITECTURE

- Can we question reality?
- We can certainly answer some questions about it
- If **λ** is a query performed about reality at time **t**
- And **c** is the time it takes to answer it
- Then a question asked at time **t** is answered by time **(t+c)**
- The system is said to have a latency of **c**

$$\lambda_t = M(R_{t-c})$$

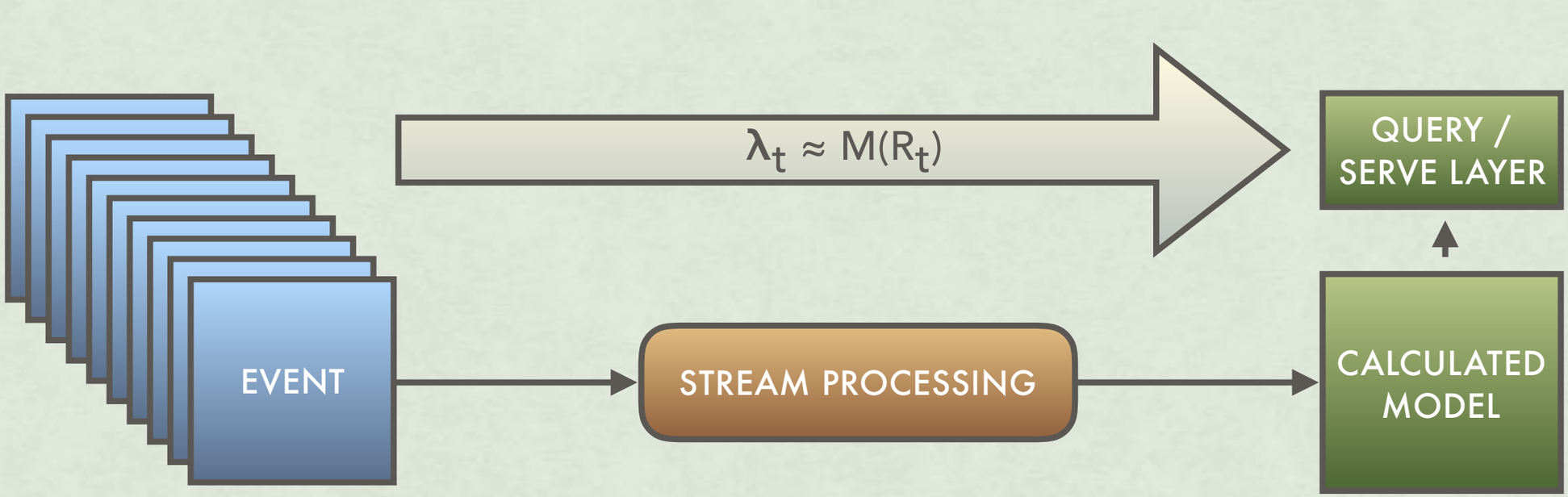
CONSTRUCTING REALITY

DECONSTRUCTING THE CULT OF CONSISTENCY

LOG BASED MICRO-SERVICES CONSTRUCT REALITY FROM EVENTS REAL-TIME

EACH EVENT IS APPLIED TO EXISTING NOTION OF REALITY IMMEDIATELY

LATENCY OR **C** TENDS TO BE VERY, VERY SMALL



MANAGING PERCEPTION

MINIMIZING C WHILE MAINTAINING A FACADE OF CONSISTENCY

- Route related events to same node(s) - ship code vs data
- Construct or update model of reality locally (in memory)
- Reconstruct state from log for recovery & change management,
- Mechanical Sympathy
    - Minimize/avoid locking, synchronization and context switching
    - CAS and FAA machine instructions plus double buffering when state must be shared
    - Treat each node as a distributed system in of itself - Cores within a socket are nodes, CPU Sockets can be thought of as racks within a data center

THE CULT OF CONSISTENCY

VS COMMAND QUERY RESPONSIBILITY SEGREGATION

- Micro-services offer encapsulation by virtue of their interface
- This interface can be a request to query its current state
- Or, a request to mutate the state managed by this micro service
- Mutating requests act as **Commands** which trigger state change
- All others requests are considered point in time snapshot **Queries**
- This pattern allows interface design that mirrors the domain model
- The persistence model is a log of all business events that occur
- State is a view derived from this log & can be reconstructed on-demand

DEMO: TIK TOK

DOMAIN, INTERFACE AND DATA MODEL

- Popular concerts are in finite supply and exhaust rapidly
- For ticket sellers the domain has these elements of interest
    - Seats during a given performance
    - Customers interested in those seats
    - Purchases that tie a customer to a seat via funding
- These form part of the domain model for this system
- They also form the basis of interfaces we expose
- And the data model is merely a log of business events

DEMO: TIK TOK

HELL FREEZES OVER