Leveraging “Right Now”:

Concepts, Challenges, and Direction for Analytics on the Wire

S. Ryan Quick *@phaedo*,
Principal Architect
PayPal Advanced Technology Group

The Wire as a “Data Space”

“Play it where it lies…”

The Wire as a "Data Space"

… and the least-utilized at that.

- We've entered the Zetabyte Era of computing. 34.9TB/sec (1.1ZB/yr) in flight on the internet at any moment. —This is only a fraction of what's moving on enterprise, scientific, academic, government networks as well.
- We understand in-situ, and are maturing at moving data to get there.
- But why wait? **The data is already there** — we just have a hard time leveraging it.

http://www.livescience.com/54094-how-big-is-the-internet.html#sthash.FpdfLuut.dpuf

The Wire as a “Data Space”

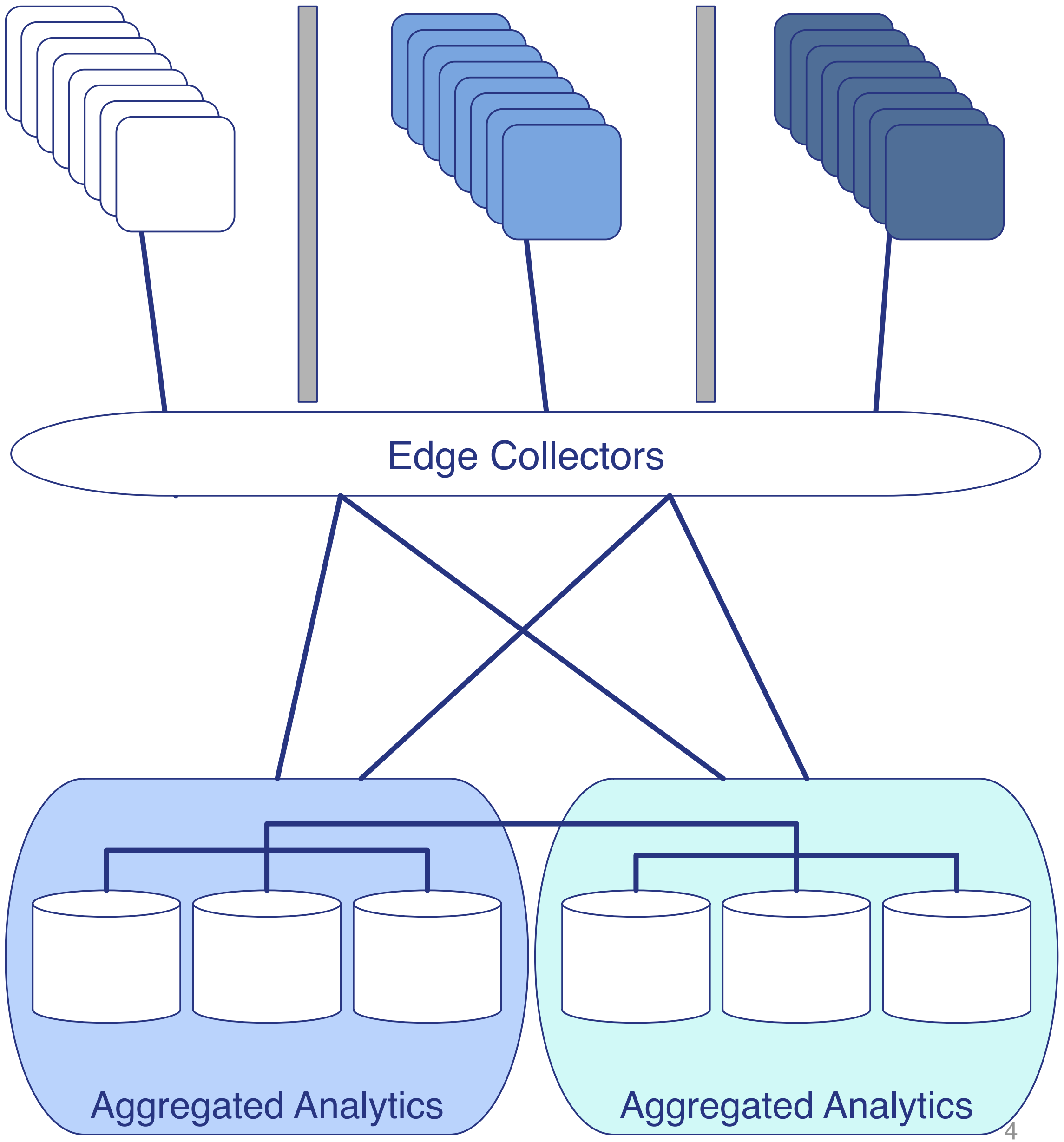
… and the least-utilized at that.

- We’ve entered the Zetabyte Era of computing. 34.9TB/sec (1.1ZB/yr) in flight on the internet at any moment. —This is only a fraction of what’s moving on enterprise, scientific, academic, government networks as well.
- We understand in-situ, and are maturing at moving data to get there.
- But why wait? **The data is already there** — we just have a hard time leveraging it.

http://www.livescience.com/54094-how-big-is-the-internet.html#sthash.FpdfLuut.dpuf

The Wire as a “Data Space”

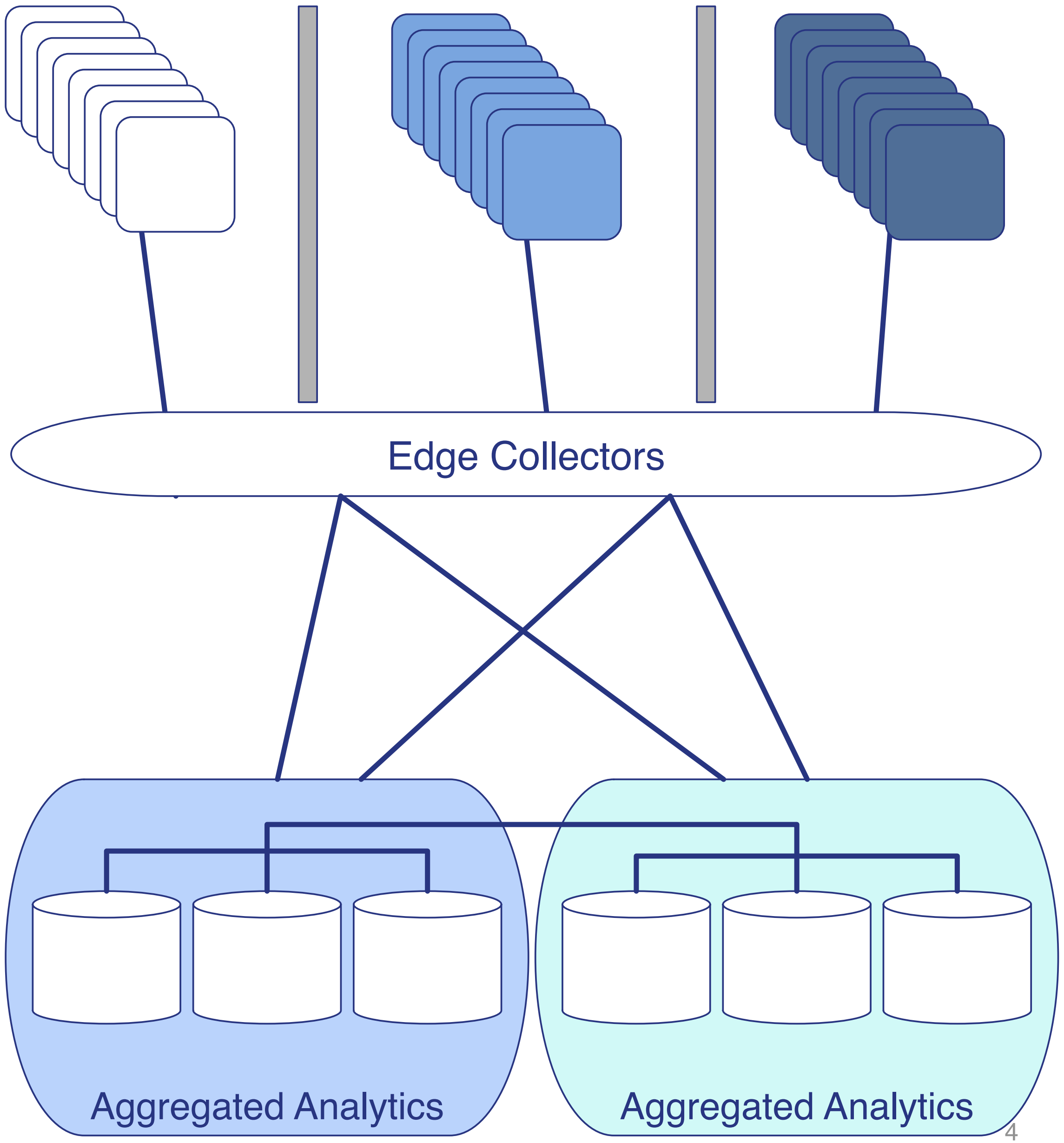
Win by **acting** on the most information.



The Wire as a “Data Space”

Win by **acting** on the most information.

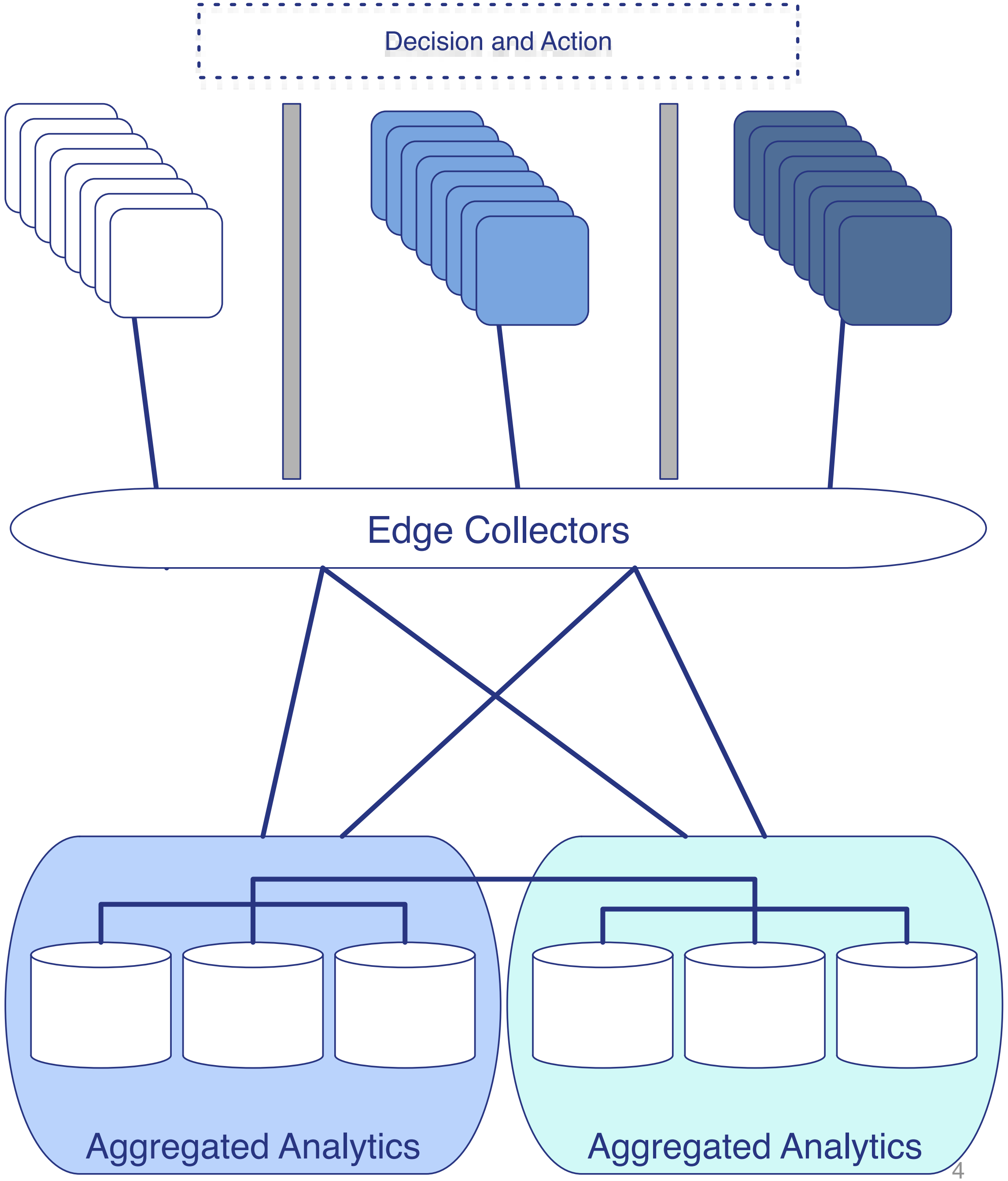
- Decisioning is fueled by information.
- As we grow more information from more sources informs better decisions — as long as we can actually handle the growth itself. That’s the crux of the problem.
- Current paradigm is to bring data to centralized systems for analysis.
- Analytic complexity directly relates to
  - distance (time, space),
  - size (atomic, chunk, overall), and
  - rate (bandwidth, throughput)



The Wire as a “Data Space”

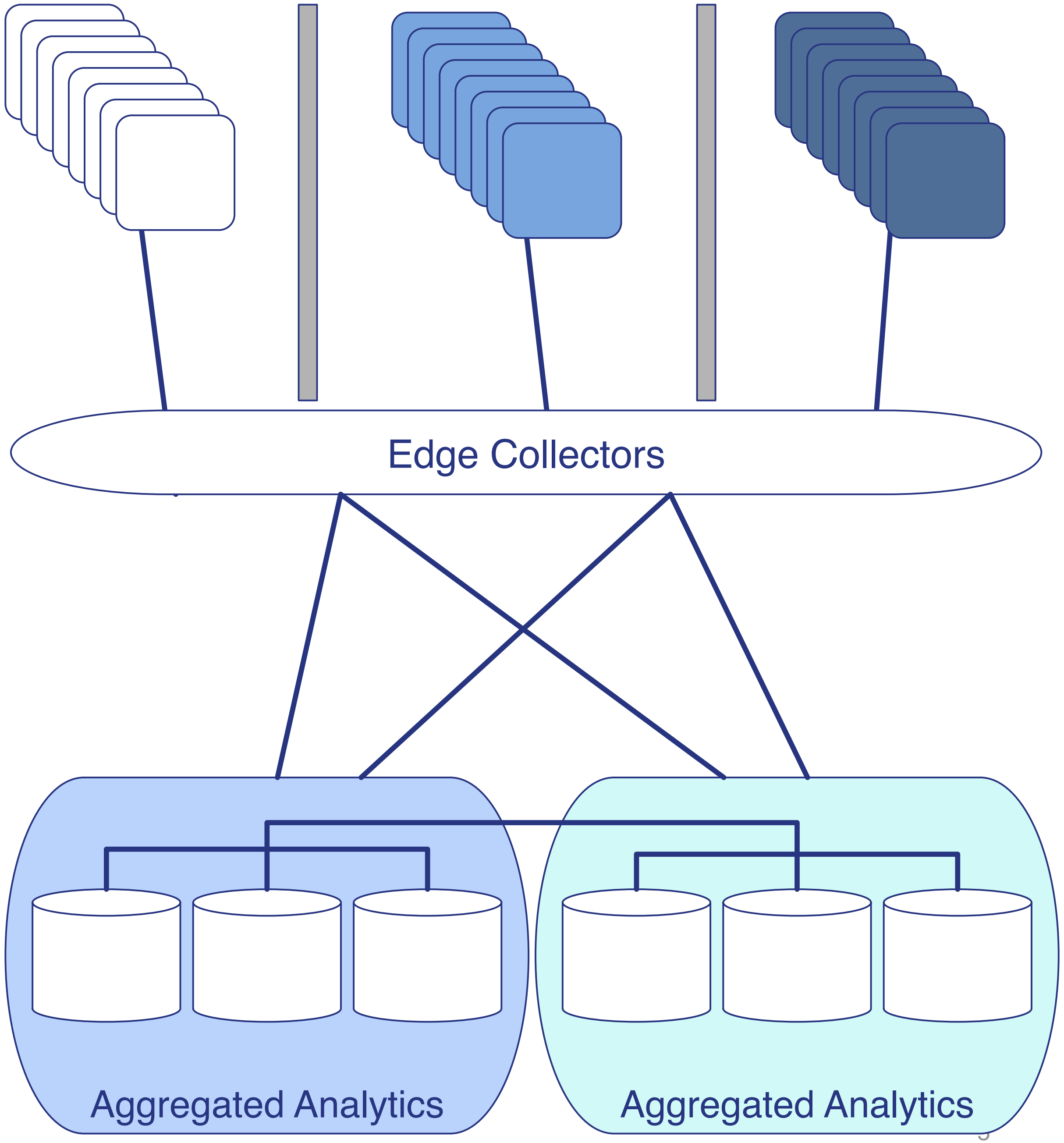
Win by **acting** on the most information.

- Decisioning is fueled by information.
- As we grow more information from more sources informs better decisions — as long as we can actually handle the growth itself. That’s the crux of the problem.
- Current paradigm is to bring data to centralized systems for analysis.
- Analytic complexity directly relates to
  - distance (time, space),
  - size (atomic, chunk, overall), and
  - rate (bandwidth, throughput)
- Decisioning, Reaction, Prediction, etc. needed at the edge — ever-increasing demand for real-time action, which necessitates real-time insight.



The Wire as a "Data Space"

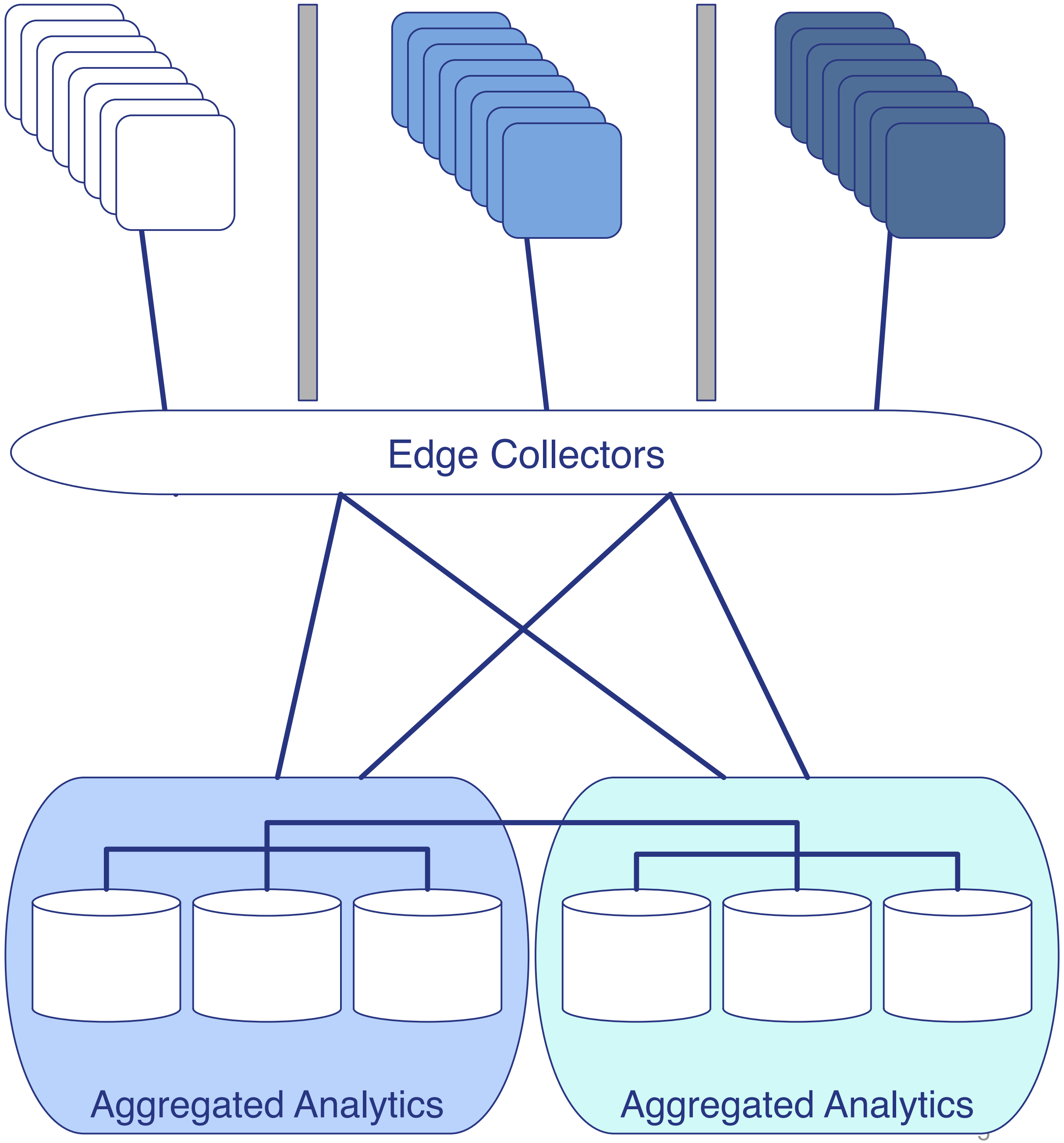
But wait! I Need All The Data!



The Wire as a "Data Space"

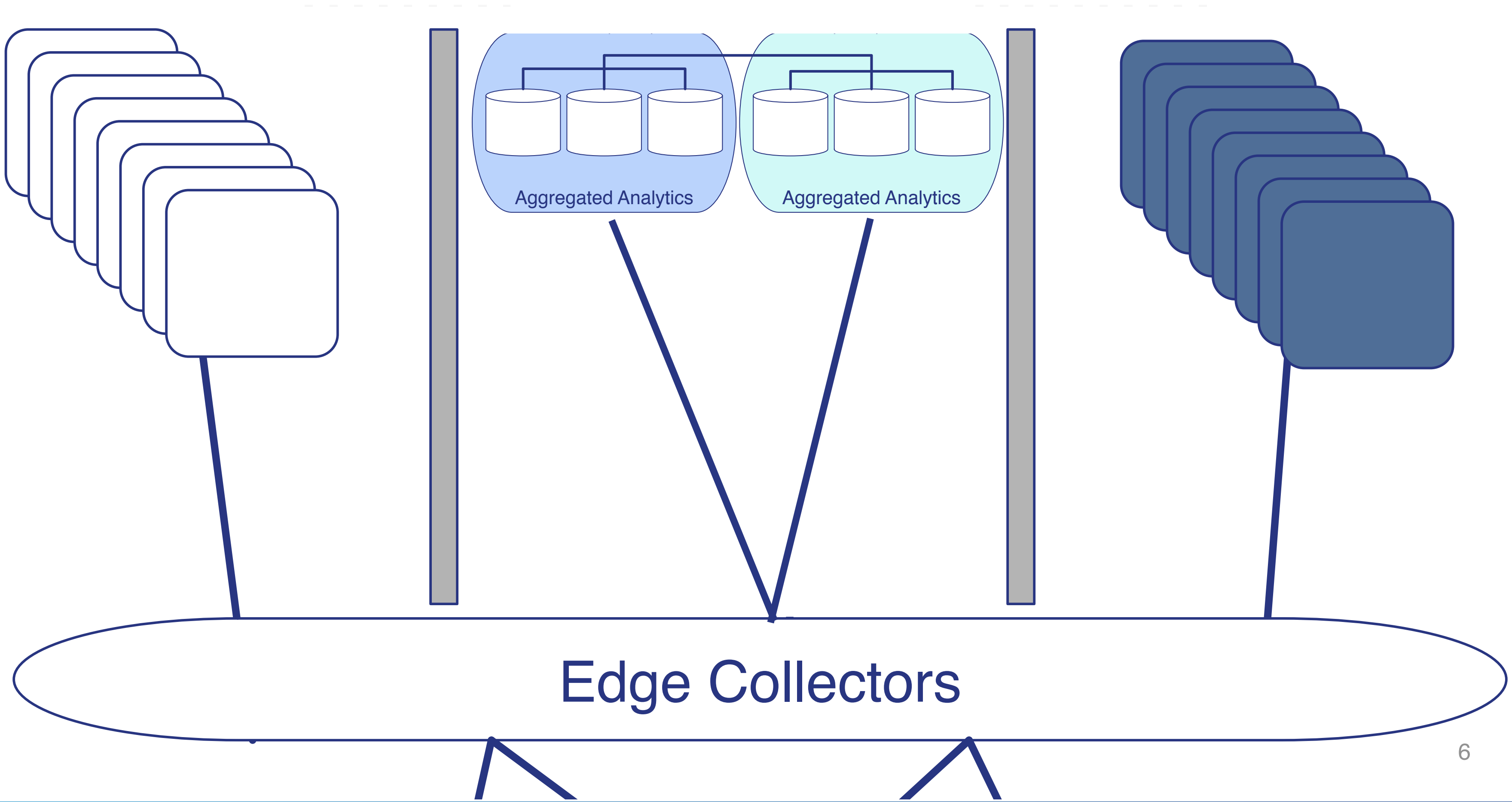
But wait! I Need All The Data!

- I need the entire dataset, from all sources, to derive information in the first place.
- My output is useful to me, but someone else will need all of the data to do their work as well.
- *(While I probably disagree, don't worry…)*



The Wire as a “Data Space”

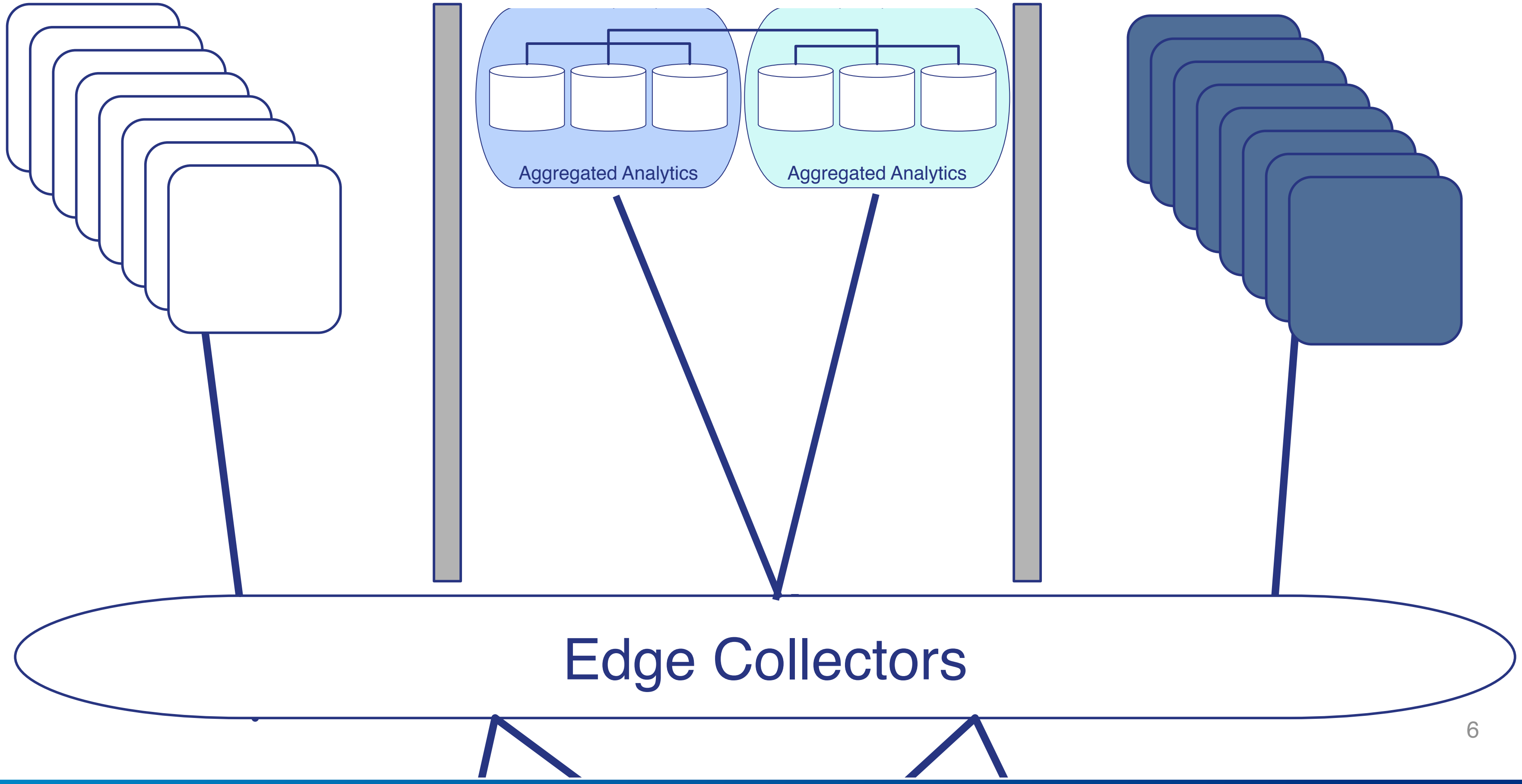
Output/Analysis is just another source…



The Wire as a "Data Space"

Output/Analysis is just another source…

- Sources can be simple, complex, small, or Big
- To leverage in-transit data, we must think beyond our use of content.
- Separate Insight from Information.
- Publish Everything.
- Let consumers consume.



Data Spaces

in-situ data at rest	***in-transit*** data moving between endpoints	***in-transform*** data under manipulation
Consistent Durable Accessible Atomic Ordered Structured (yes, even “unstructured data”)	Consistent* Transient*/Durable* Accessible Atomic*/Parallel* Ordered* Structured	Consistent* Transient/Durable* Accessible* Atomic Ordered Structured
Single data access Multichannel delivery*	Multichannel data access* Multichannel delivery*	Single data access† Single channel delivery
Commonly called “data at rest”	Data “in flight” or moving between endpoints	Data active manipulation (augmentation, transformation, reduction, format alteration, etc.)

* Configurable, depending on capability/need   † This is changing w/ new hardware options/implementations

Data Spaces

in-situ data at rest	***in-transit*** data moving between endpoints	***in-transform*** data under manipulation
Consistent Durable Accessible Atomic Ordered Structured (yes, even "unstructured data")	Consistent* Transient*/Durable* Accessible Atomic*/Parallel* Ordered* Structured	Consistent* Transient/Durable* Accessible* Atomic Ordered Structured
Single data access Multichannel delivery*	Multichannel data access* Multichannel delivery*	Single data access† Single channel delivery
Commonly called "data at rest"	Data "in flight" or moving between endpoints	Data active manipulation (augmentation, transformation, reduction, format alteration, etc.)

* Configurable, depending on capability/need † This is changing w/ new hardware options/implementations

In-Transit Technology Concepts

"Sphere of Influence": Transmission does not just relay information, but orders and gives meaning to it — increasing both insight and information itself.

In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Time

- Ordering
  - (in-order, out-of-order, random, reverse, delayed)

Space

- Freshness
- Observational Distance
- Decay

Magnitude

- Object
- Information
- Data
- Relevance

In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Time

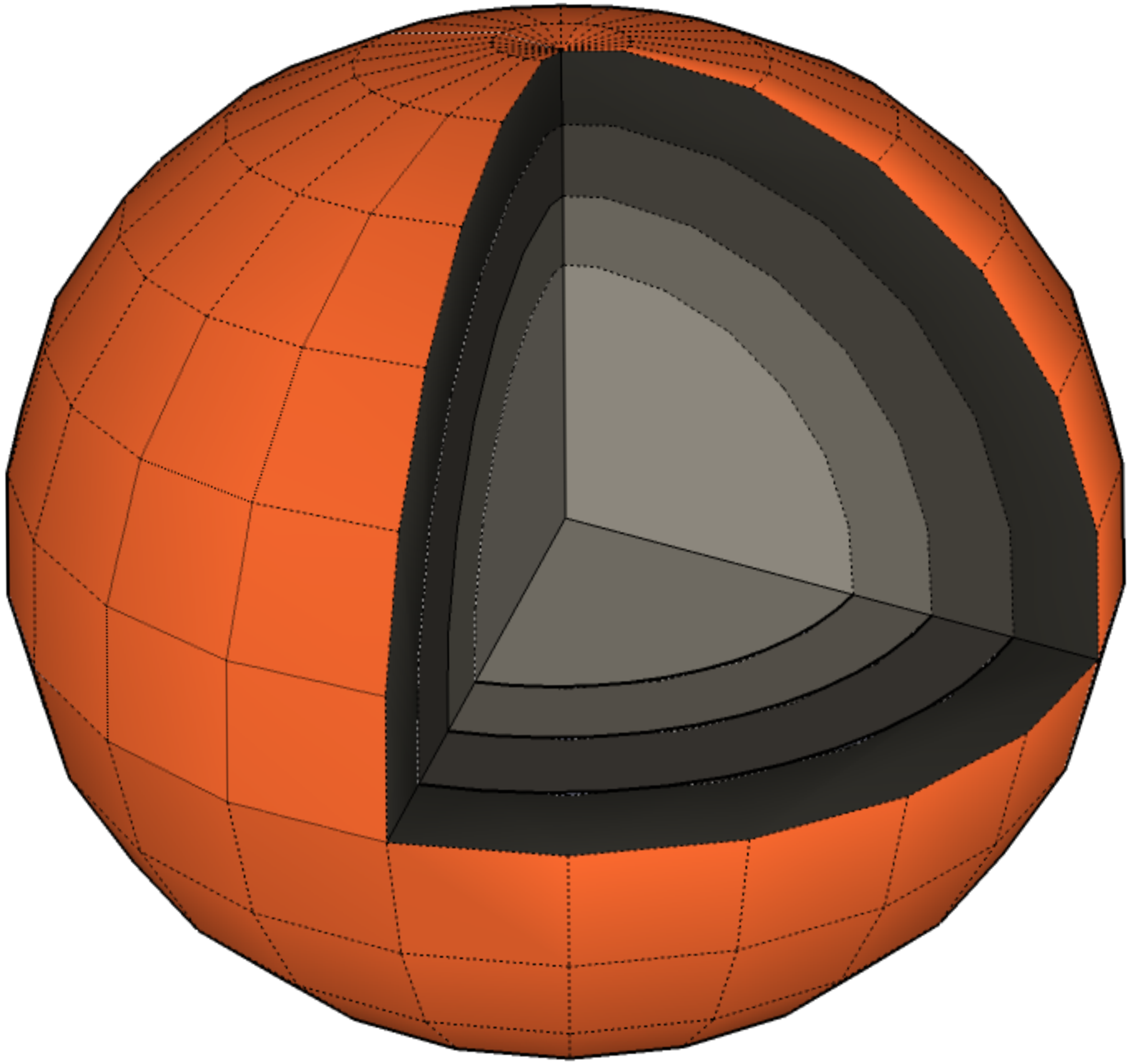
- Ordering
  - (in-order, out-of-order, random, reverse, delayed)

Space

- Freshness
- Observational Distance
- Decay

Magnitude

- Object
- Information
- Data
- Relevance



In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Time

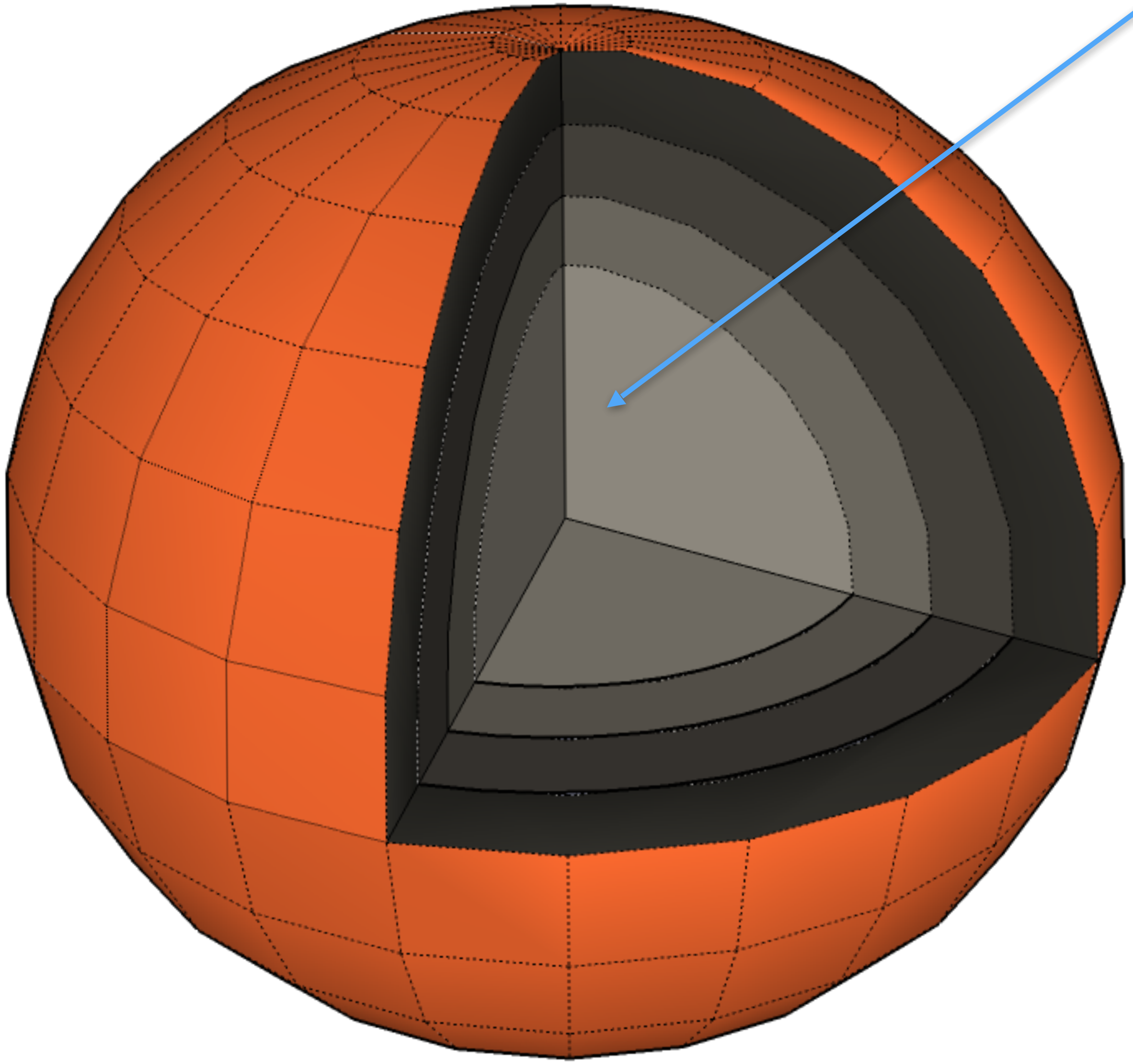
- Ordering
  - (in-order, out-of-order, random, reverse, delayed)

Space

- Freshness
- Observational Distance
- Decay

Magnitude

- Object
- Information
- Data
- Relevance



T0

Earthquake initial event, no data

In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Time

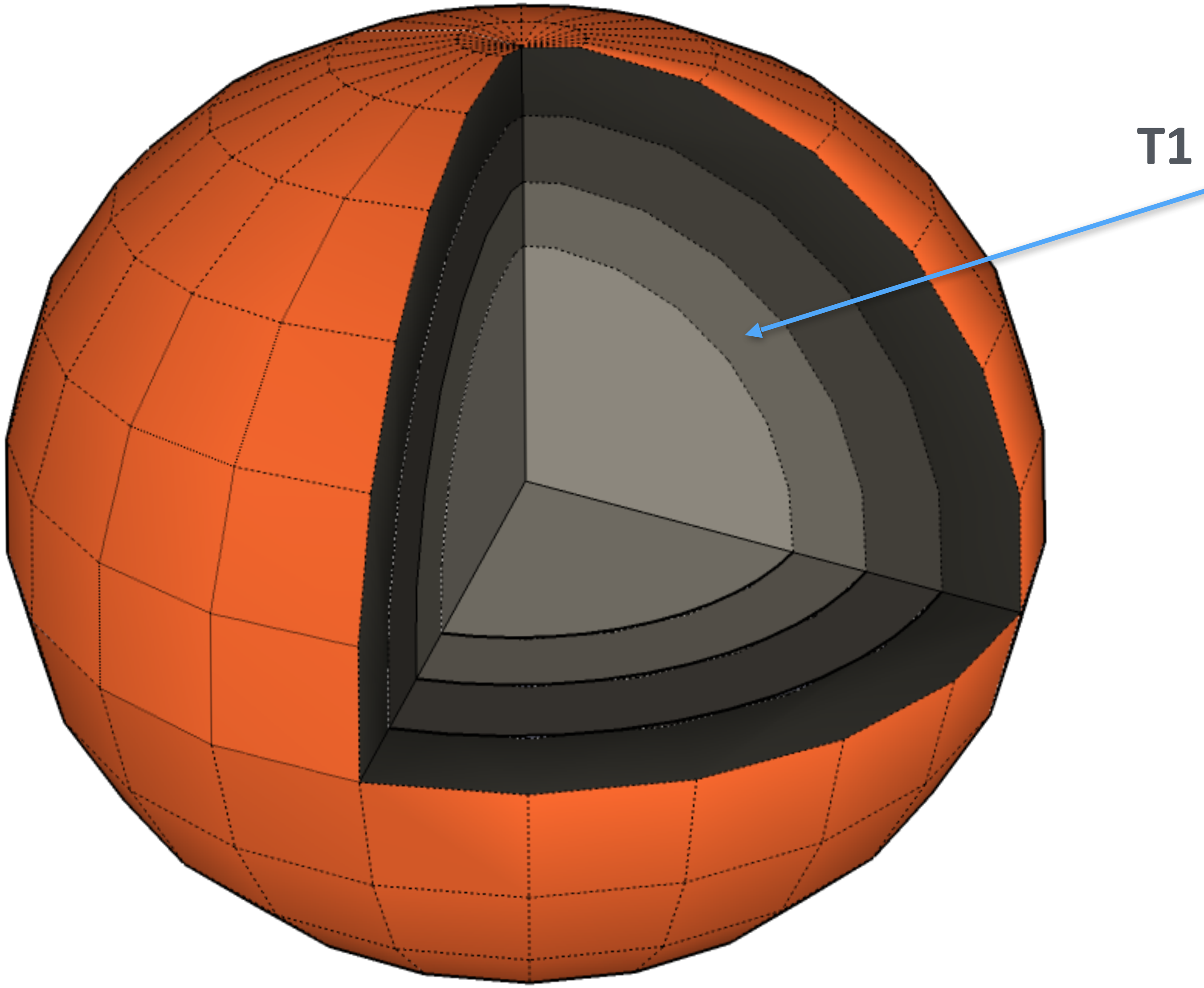
- Ordering
  - (in-order, out-of-order, random, reverse, delayed)

Space

- Freshness
- Observational Distance
- Decay

Magnitude

- Object
- Information
- Data
- Relevance



Seismometer, initial data generated about event

PayPal

In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Time

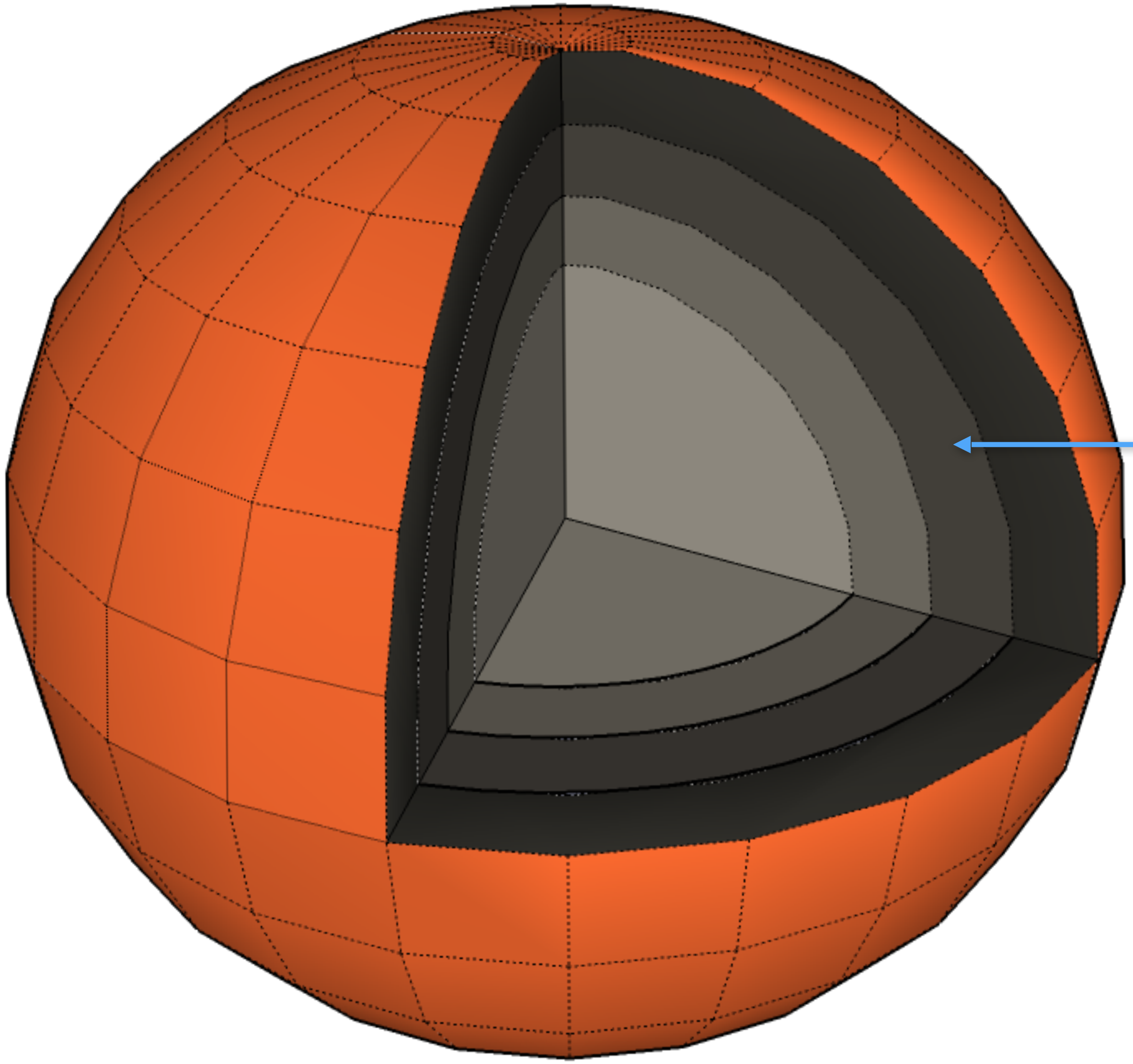
- Ordering
  - (in-order, out-of-order, random, reverse, delayed)

Space

- Freshness
- Observational Distance
- Decay

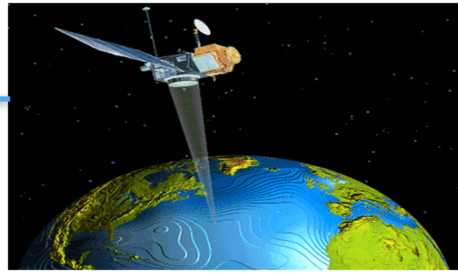
Magnitude

- Object
- Information
- Data
- Relevance



Satellite Receiver data transformed to message

T2



In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Time

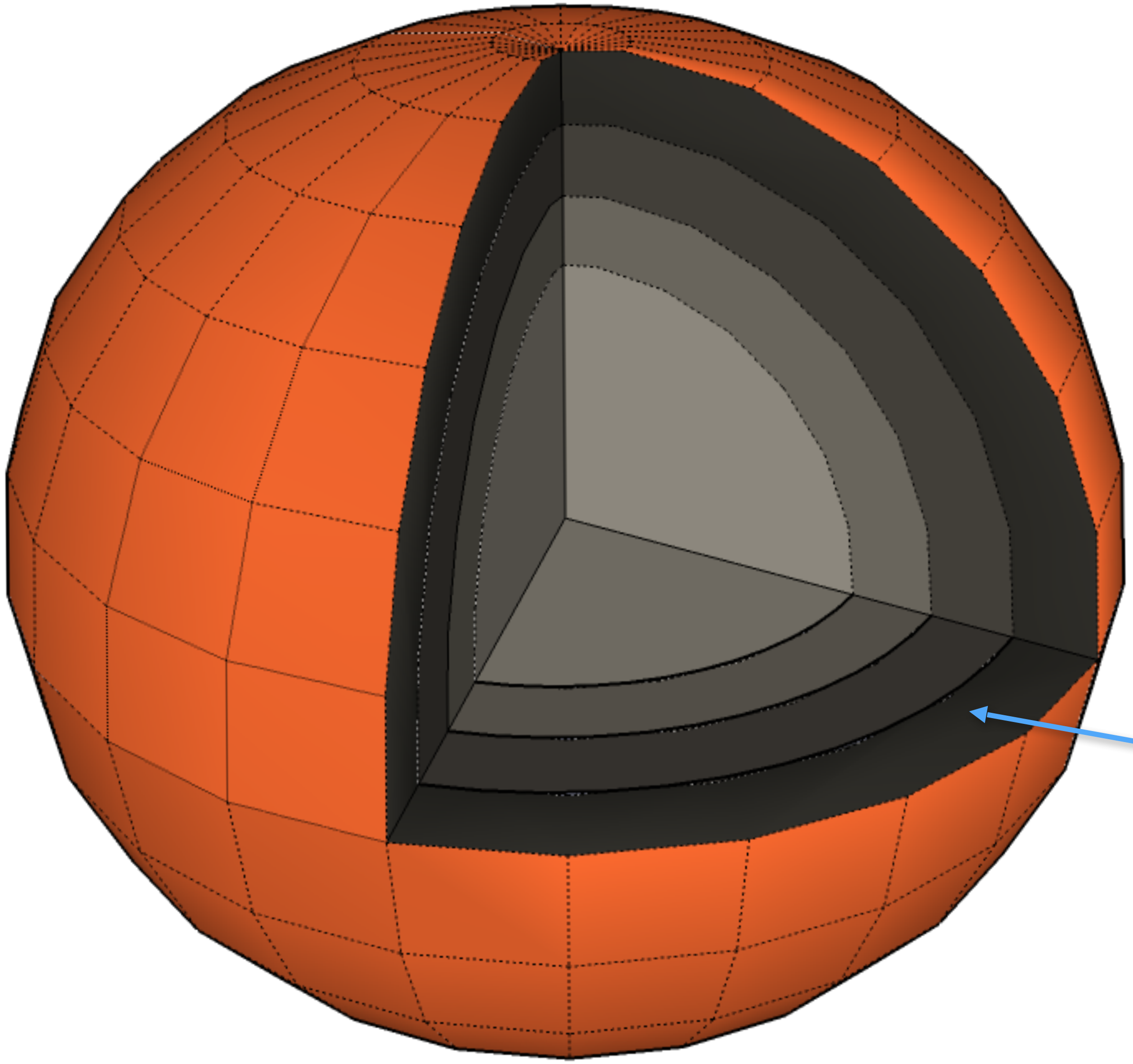
- Ordering
  - (in-order, out-of-order, random, reverse, delayed)

Space

- Freshness
- Observational Distance
- Decay

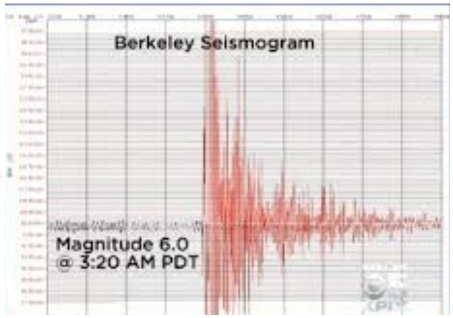
Magnitude

- Object
- Information
- Data
- Relevance



Ground station reception message transformed to human readable format

T3



In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Time

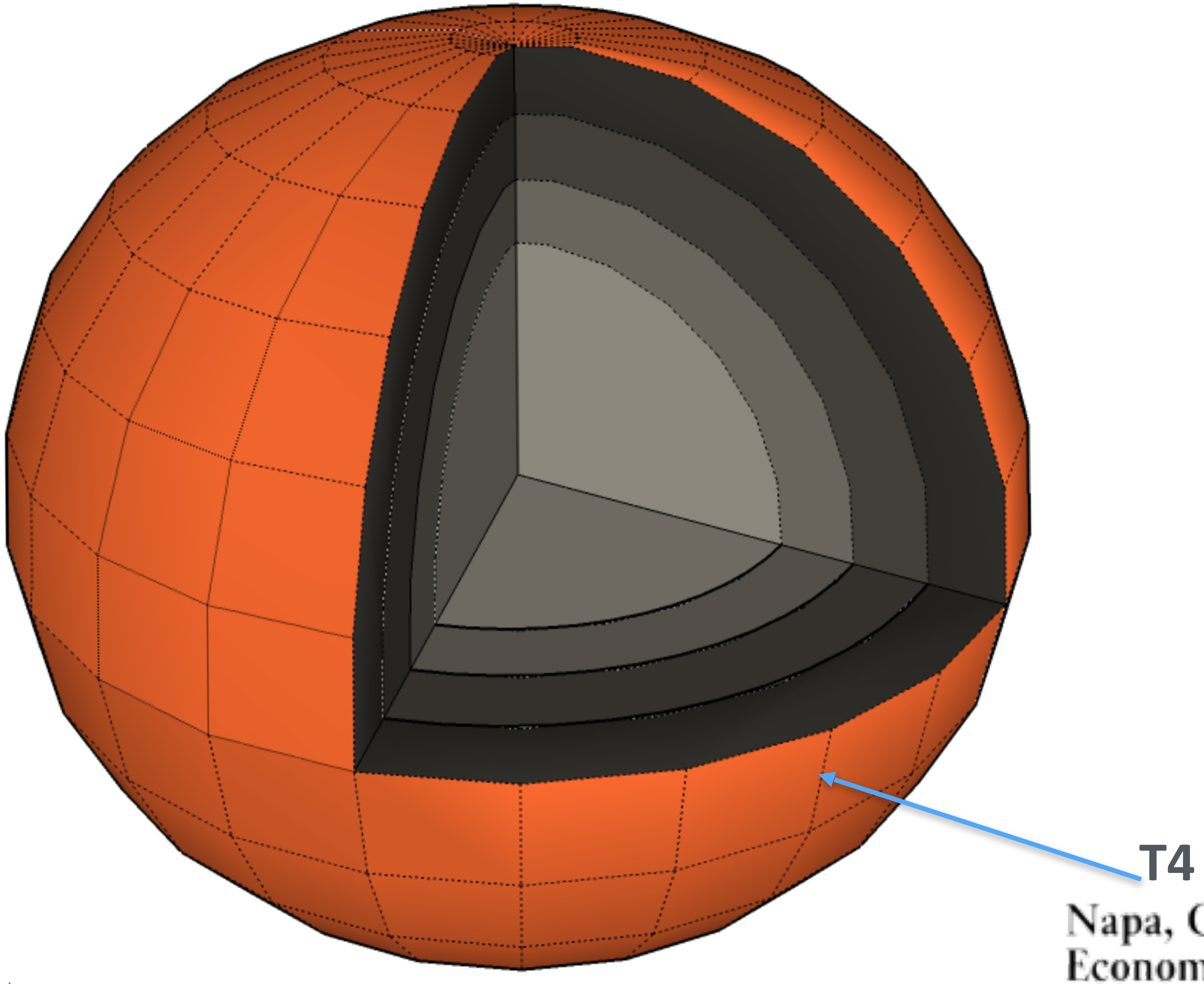
- Ordering
  - (in-order, out-of-order, random, reverse, delayed)

Space

- Freshness
- Observational Distance
- Decay

Magnitude

- Object
- Information
- Data
- Relevance



Economic impact post-event data

Napa, Calif., earthquake: Economic hit could reach $1 billion

@phaedo

In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Time

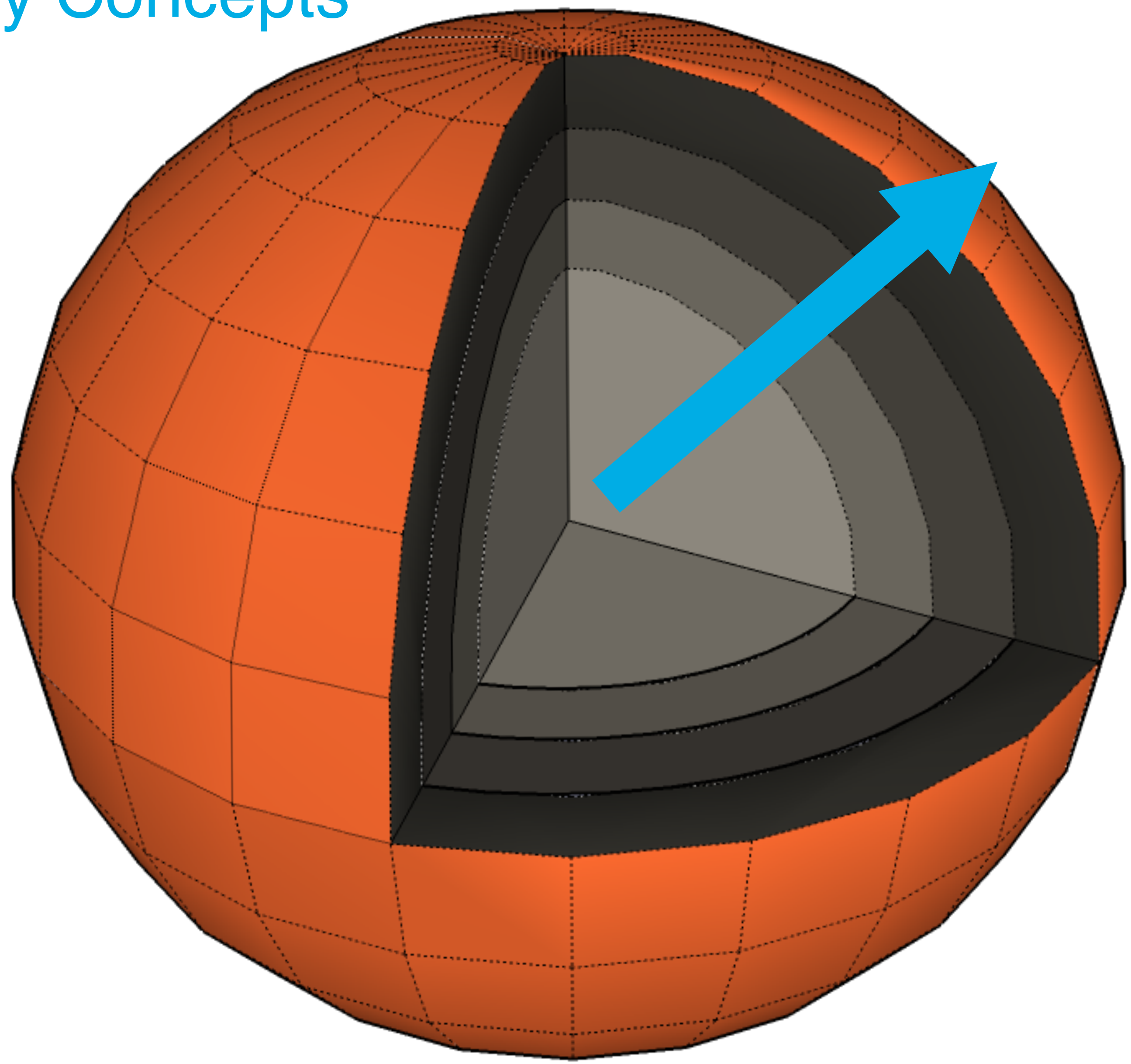
- **Ordering**
  - (**in-order**, out-of-order, random, reverse, delayed)

Space

- Freshness
- Observational Distance
- Decay

Magnitude

- Object
- Information
- Data
- Relevance



In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Time

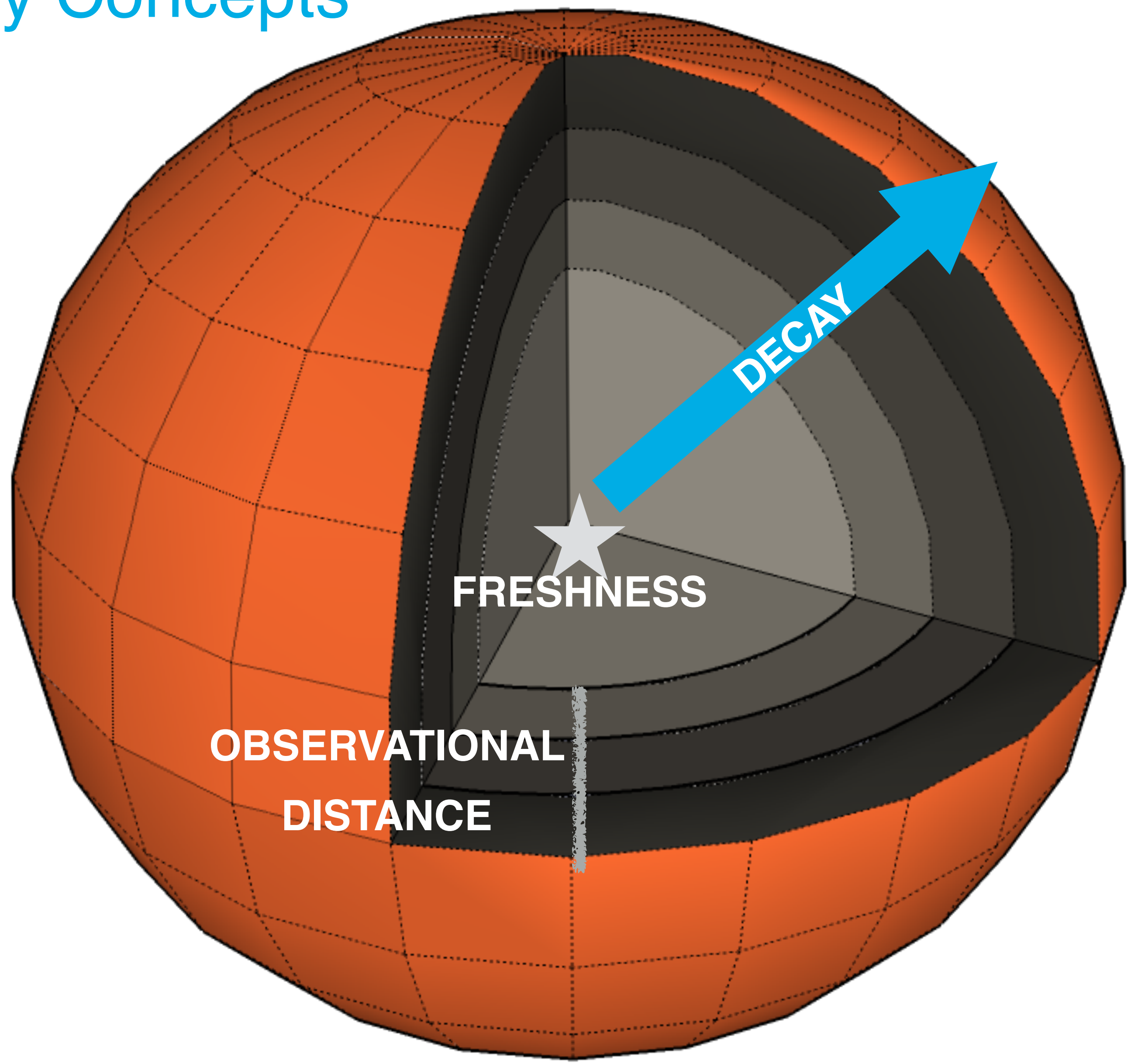
- Ordering
  - (in-order, out-of-order, random, reverse, delayed)

Space

- **Freshness**
- **Observational Distance**
- **Decay**

Magnitude

- Object
- Information
- Data
- Relevance



In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Time

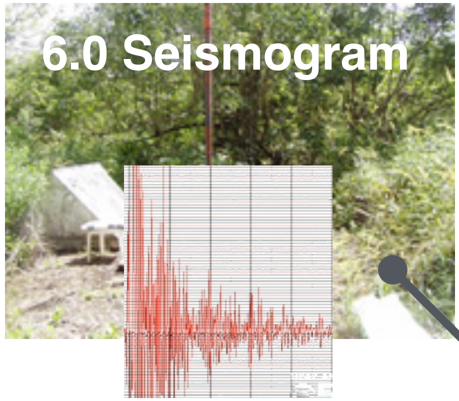
- Ordering
  - (in-order, out-of-order, random, reverse, delayed)

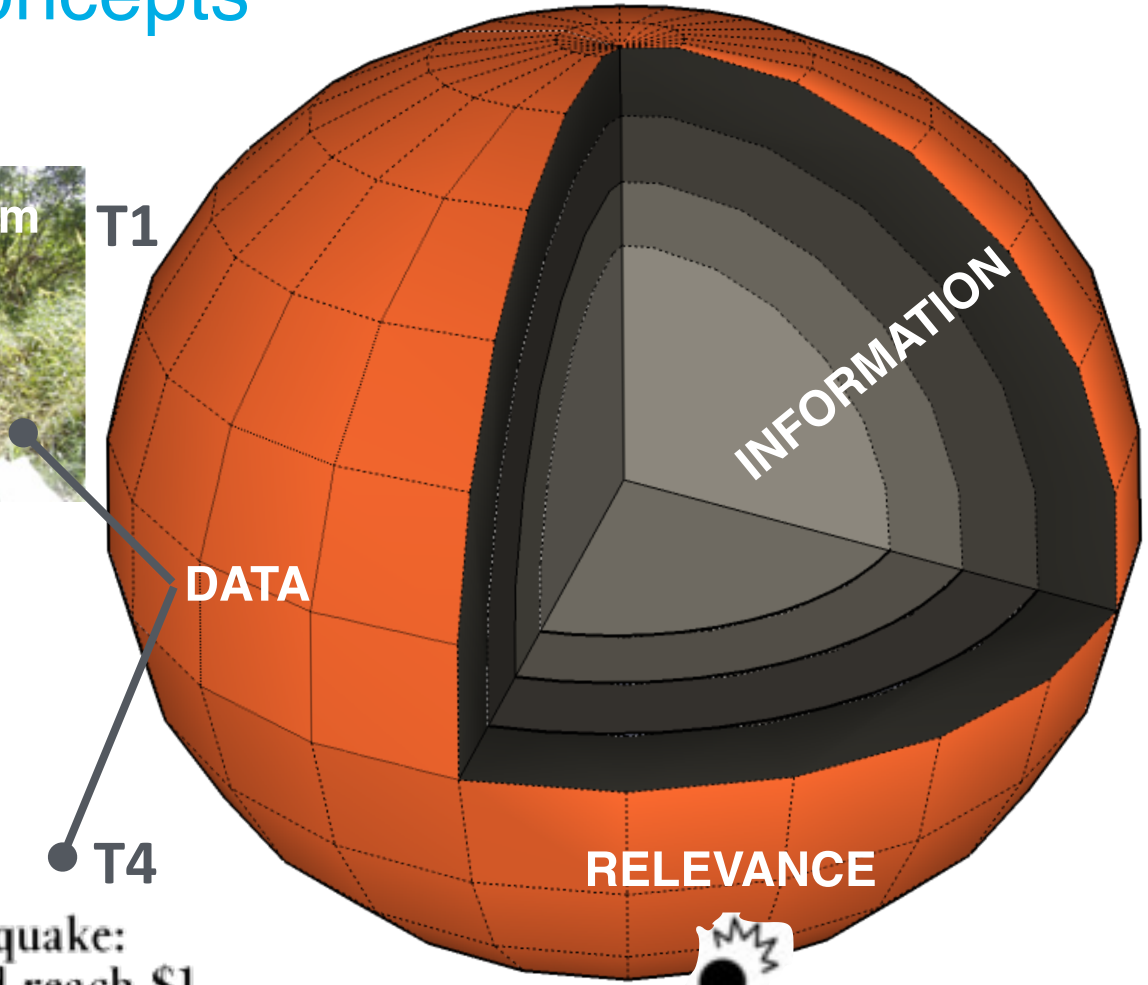
Space

- Freshness
- Observational Distance
- Decay

Magnitude

- **Object**
- **Information**
- **Data**
- **Relevance**





Napa, Calif., earthquake: Economic hit could reach $1 billion

In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Repeatability

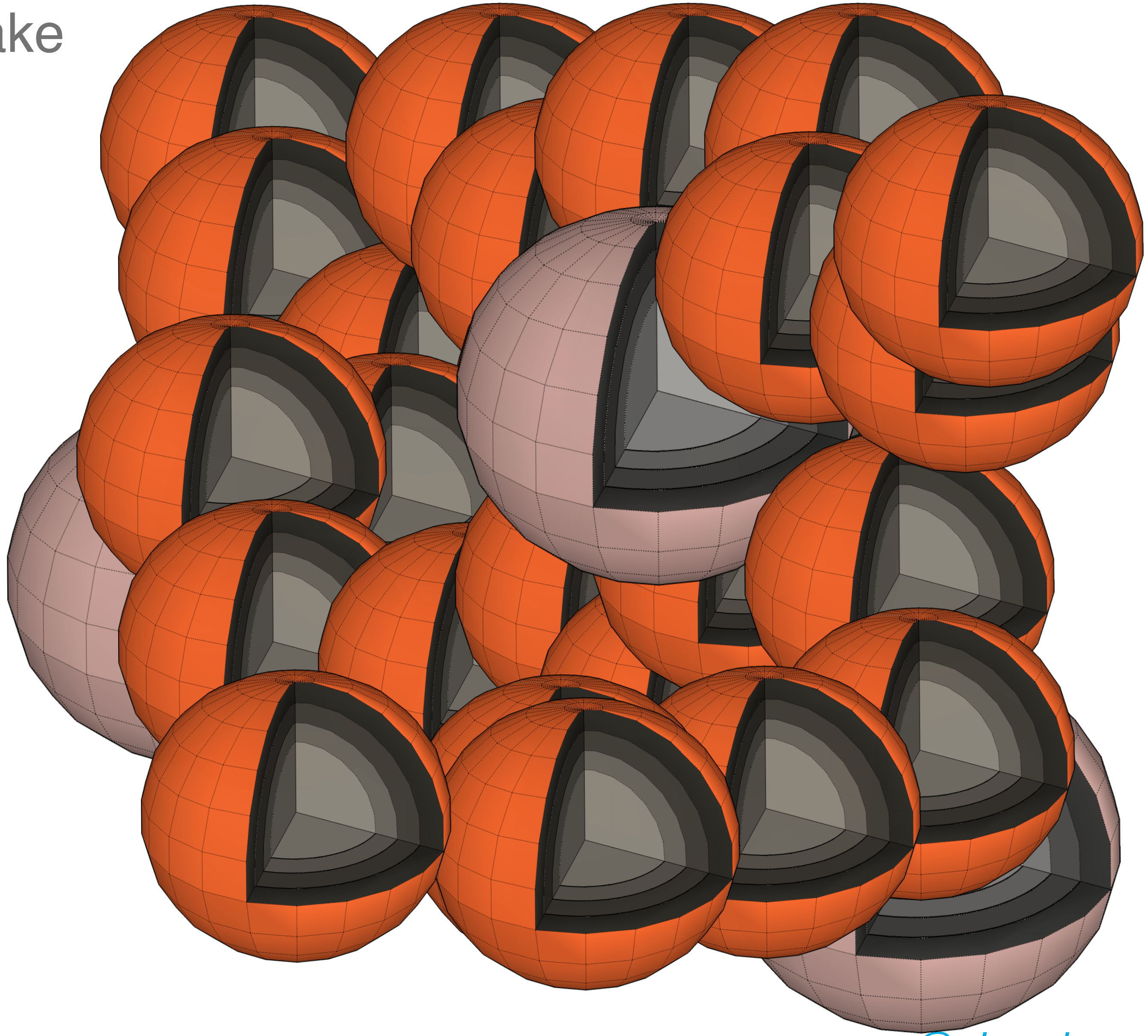
- **Duplication**
- **Replication**
- **Uniqueness**
- **Similarity**

Atomicity

- Influence
- Interdependency
- Discreteness

Longevity

- Persistence
- Retention
- Durability



In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Repeatability

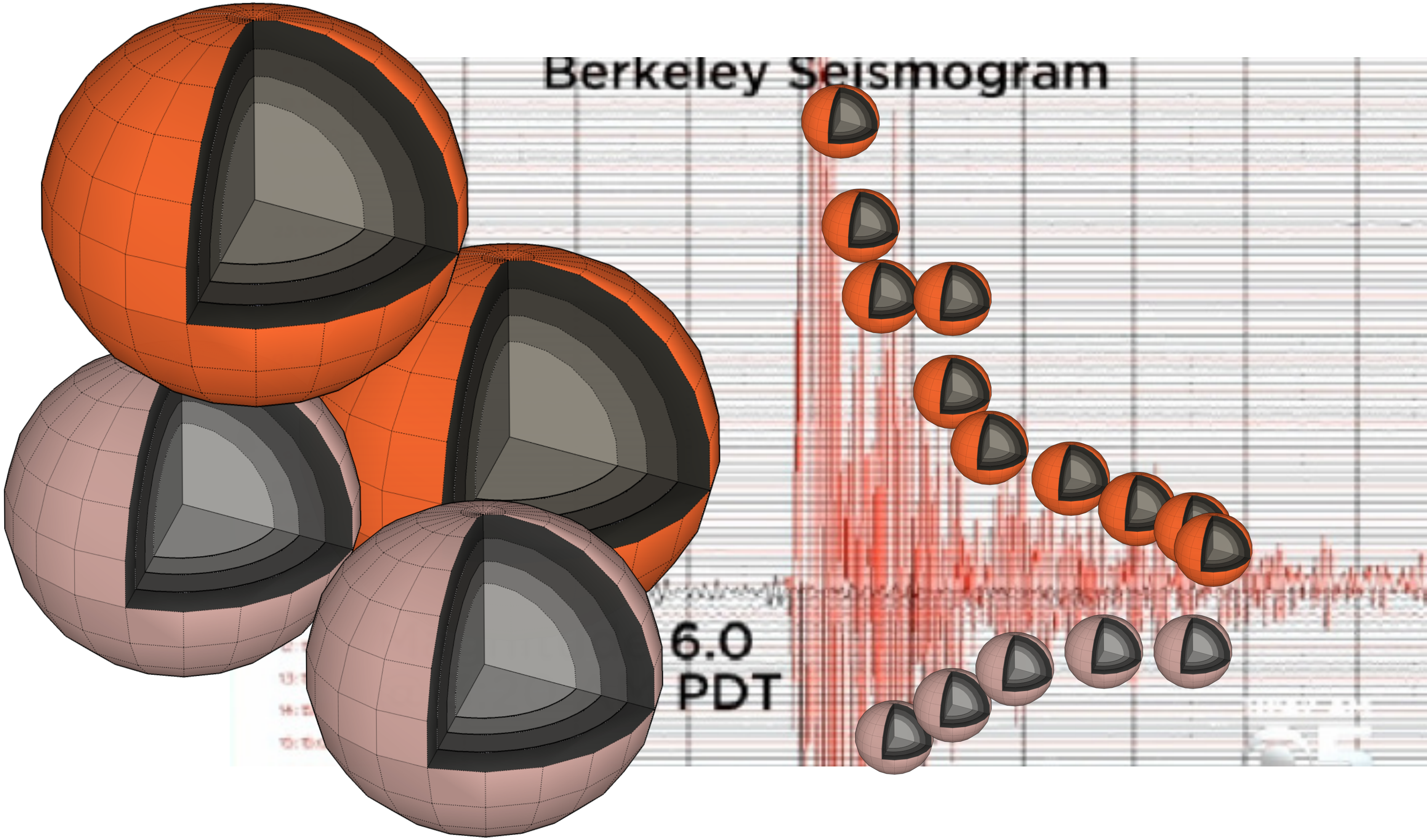
- Duplication
- Replication
- Uniqueness
- Similarity

Atomicity

- **Influence**
- **Interdependency**
- **Discreteness**

Longevity

- Persistence
- Retention
- Durability



In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Repeatability

- Duplication
- Replication
- Uniqueness
- Similarity

Atomicity

- Influence
- Interdependency
- Discreteness

Longevity

- **Persistence**
- **Retention**
- **Durability**

Date	August 24, 2014
Origin time	10:20:44 UTC[1]
Magnitude	6.0 M_w[1]
Depth	7 mi (11 km)[1]
Epicenter	38.22°N 122.31°W[1]
Fault	West Napa Fault
Type	Strike-slip[1]
Areas affected	North Bay (San Francisco Bay Area) California, United States
Total damage	$362 million–$1 billion[2][3]
Max. intensity	VIII (*Severe*)[1]
Casualties	1 killed[4] about 200 injured[5]

In-Transit — Simple Technology Concepts

Data Sphere - 2014 Napa Earthquake

Repeatability

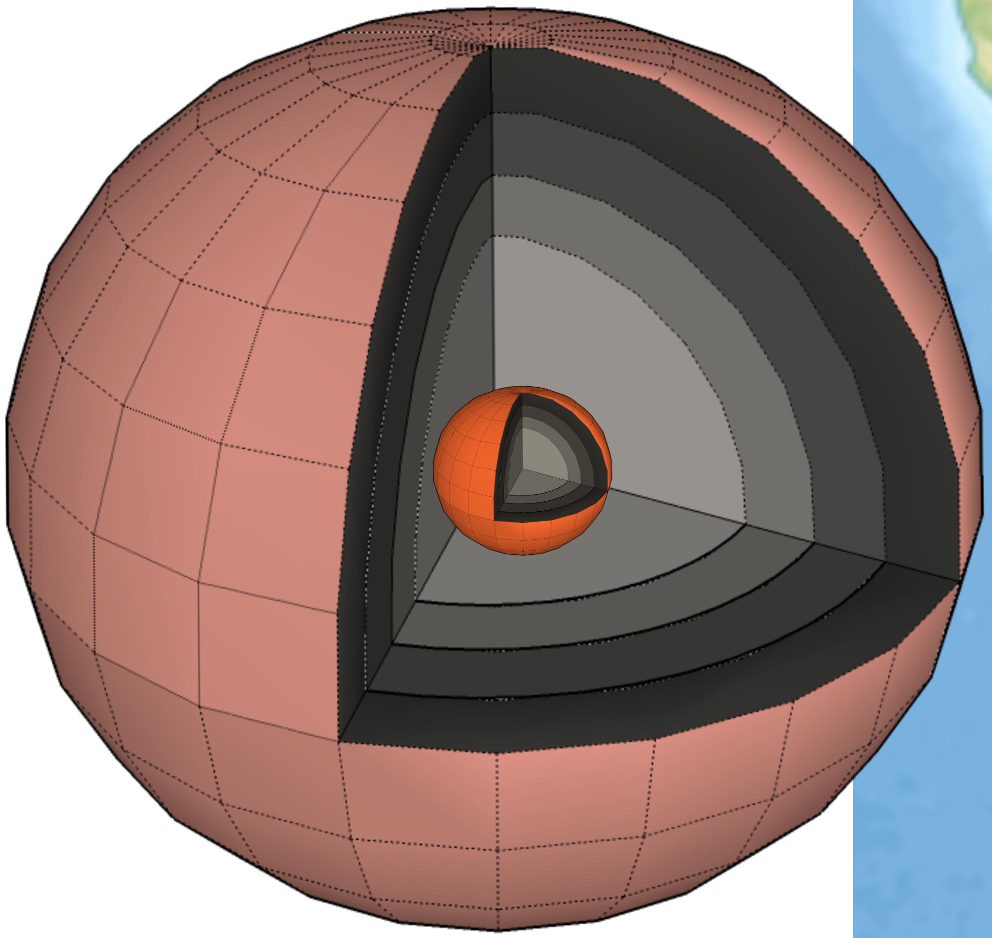
- Duplication
- Replication
- Uniqueness
- Similarity

Atomicity

- Influence
- Interdependency
- Discreteness

Longevity

- **Persistence**
- **Retention**
- **Durability**



Date	August 24, 2014
Origin time	10:20:44 UTC[1]
Magnitude	6.0 M_w[1]
Depth	7 mi (11 km)[1]
Epicenter	38.22°N 122.31°W[1]
Fault	West Napa Fault
Type	Strike-slip[1]
Areas affected	North Bay (San Francisco Bay Area) California, United States
Total damage	$362 million–$1 billion[2][3]
Max. intensity	VIII (*Severe*)[1]
Casualties	1 killed[4] about 200 injured[5]

In-Transit Technology Complex Concepts

leveraging simple concepts as building blocks

Compounded Dimensions and Series, Folded Dimensions and Series

Pattern Recognition

- Anomaly/Similarity Detection
- Frequency
- Magnitude
- Relative (Correlation/Negation/Absence)

State Change

- Event
- Observation
- Insight
- Data
- Request/Reply

In-Transit Technology Complex Concepts

leveraging simple concepts as building blocks

Compounded Dimensions and Series, Folded Dimensions and Series

Stream Manipulation

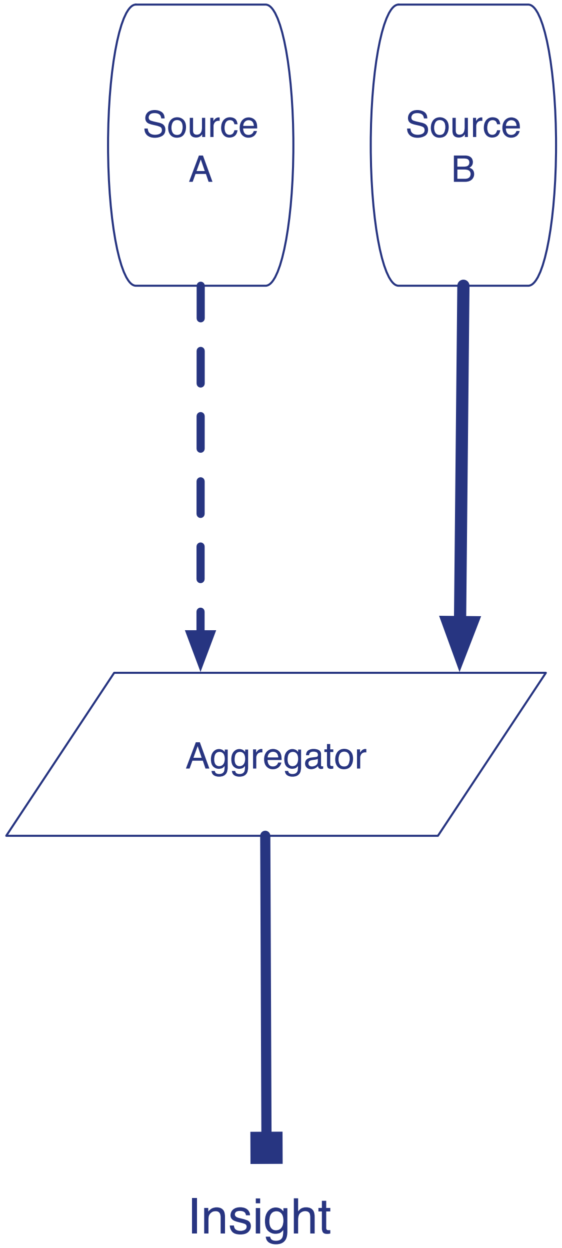
- Derivation
- Creation
- Replication
- Combination
- Views

Information and Insight

- Parallel Source
- Folded Source
- Source Augmentation

In-Transit Data Analytics Approaches

design patterns and sample use-cases



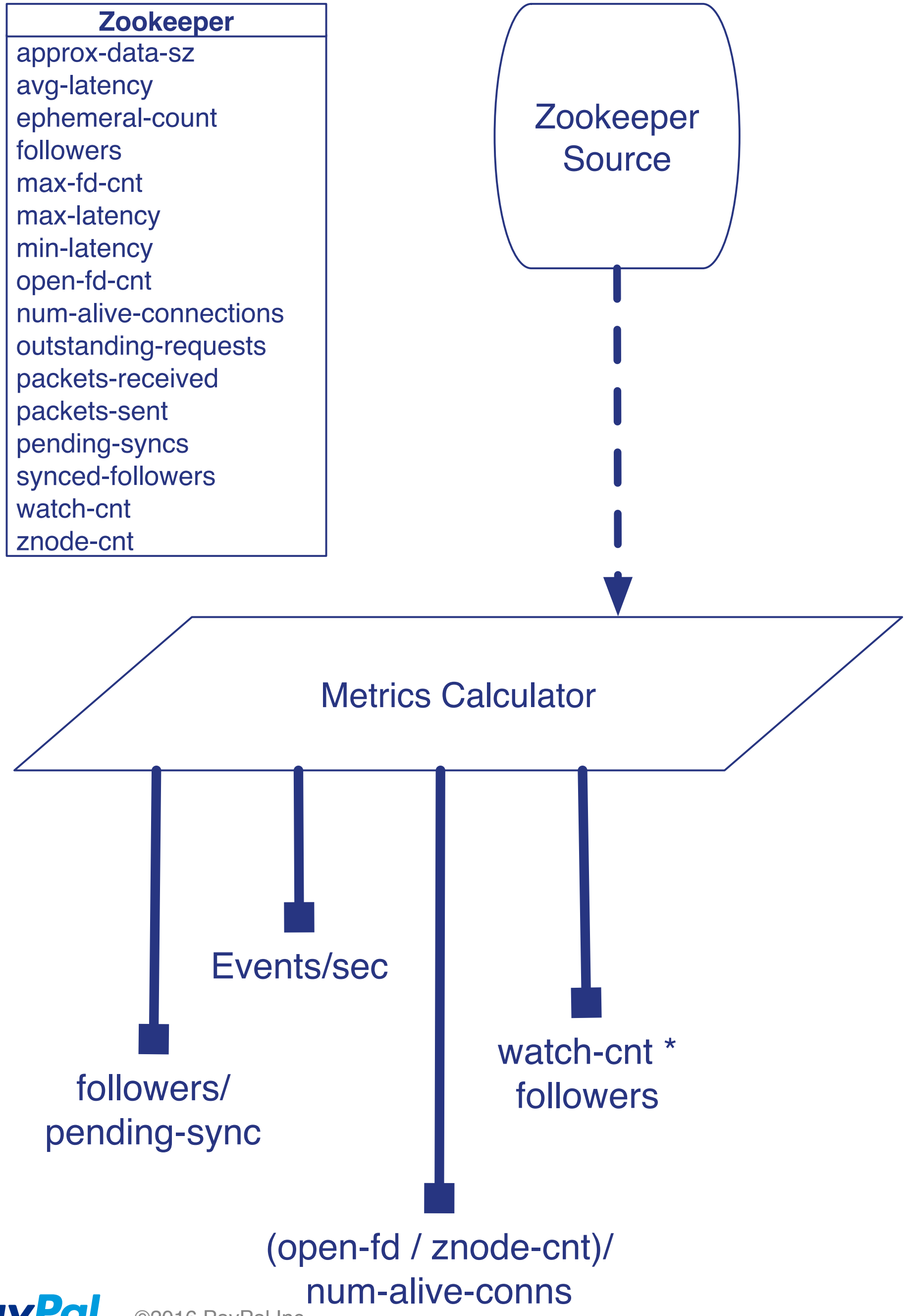
Simple Approaches

Aggregation

Event Statistics

Atomic Pattern Recognition

- Stream sources are combined in an aggregation application.
- Output is derived insight based on both sources
- **Use Case Example: CPU performance related to TCP Connections**
  - A: CPU idle % every 30s
  - B: TCP connections (incoming) [Event Driven]
  - INSIGHT: TCPCONNS/IDLE %



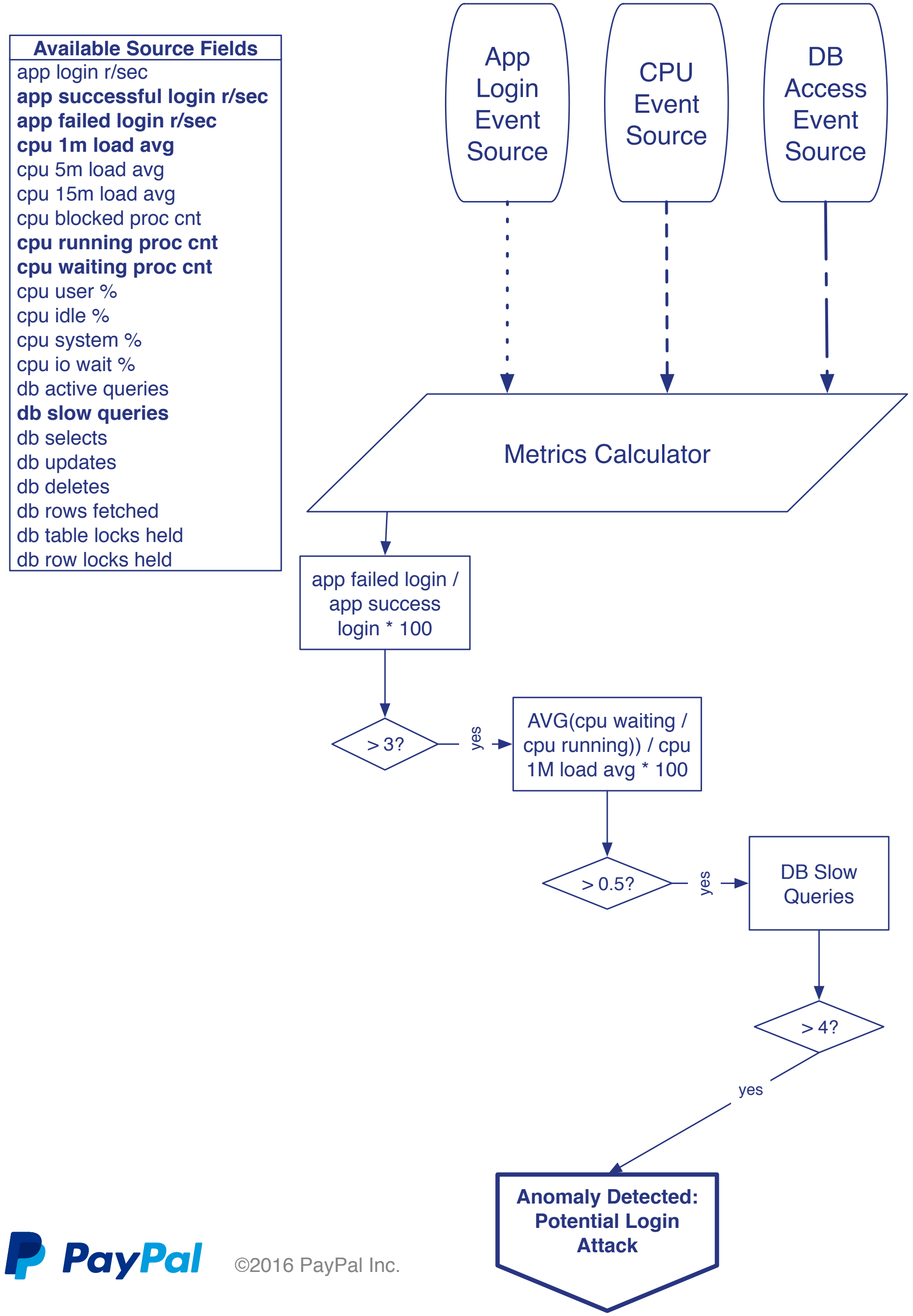
Simple Approaches

Aggregation

Event Statistics

Atomic Pattern Recognition

- Numerical/Categorial calculations based on data contained within the source datum/event
- Output insight effectively introduces new sources, generally numerical/gauged.
- **Use Case Example: Watched-Files-Per-Active-Consumer output as new stream source**
  - INSIGHT: *watch-cnt* (value per event) * *synced-followers* (value per event)



Simple Approaches

Aggregation

Event Statistics

Atomic Pattern Recognition

- Simple thresholds within the event itself
- Correlation can be within a single source, or across disparate sources
- Represented as “waterfalling” but this depends on frequency and is really just easier for us to read, the operations are parallel and stateless (in this approach)
- **Use Case Example: Output Potential-Login-Attack events**

Compound Approaches

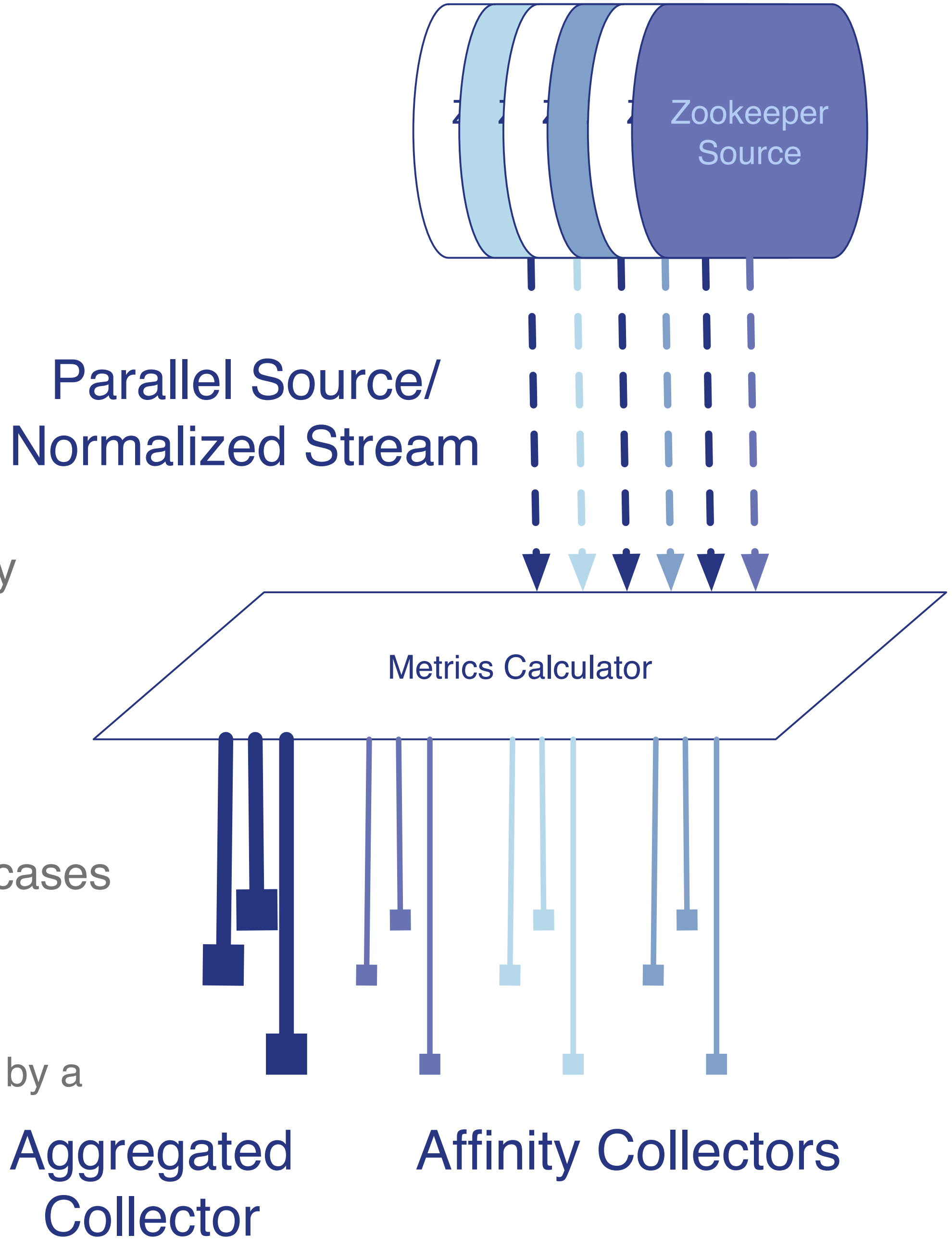
Affinity + Simple Case

Stream + Augmented Datasource

Parallel Stream

Frequency-Shifted Stream

- Given parallel publishers for single source schemas, affinity refers to collating events by
  - publisher
  - schema
  - both
- Can be implemented automatically based on other simple cases
- **Use Case Example: "Person of Interest", "Behavior of Interest"**
  - Collate data by publisher once an anomalous event is triggered by a simple approach
  - Collate all like-schema sources to watch "pool behavior"



Compound Approaches

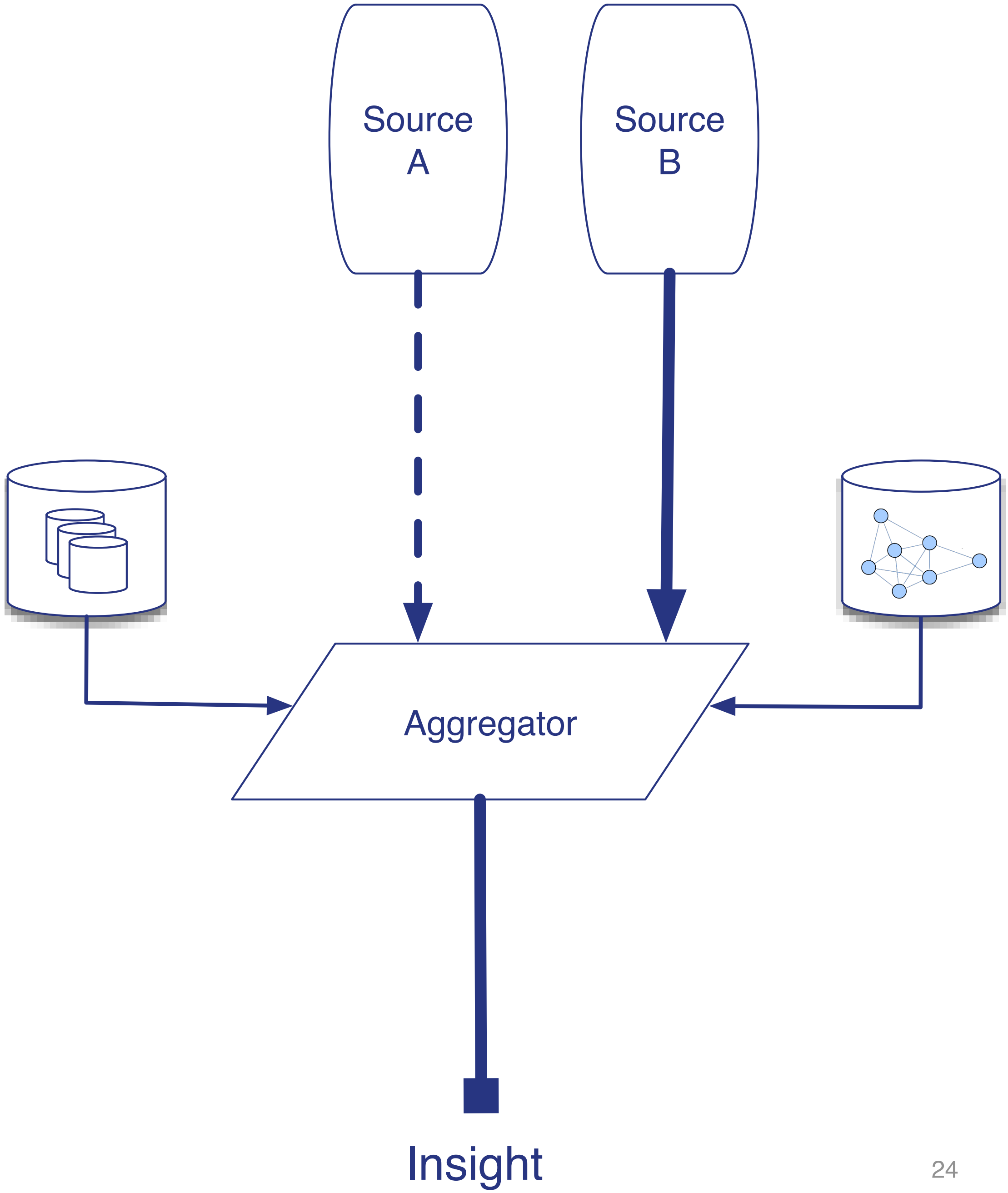
Affinity + Simple Case

Stream + Augmented Datasource

Parallel Stream

Frequency-Shifted Stream

- Source data is augmented by
  - additional sources (alternate schema)
  - additional data sources (RDBMS, GraphDB, KV, Cache, etc)
- Used in cases where information on the wire requires additional context, culling, augmentation to provide insight
- **Use Case Example: Network Detection**
  - Event Source provides transaction details, network actors
  - RDBMS provides known-network attributes
  - Graph DB provides existing actor-network
  - Aggregator determines similarity score that the current event is a particular network type



Compound Approaches

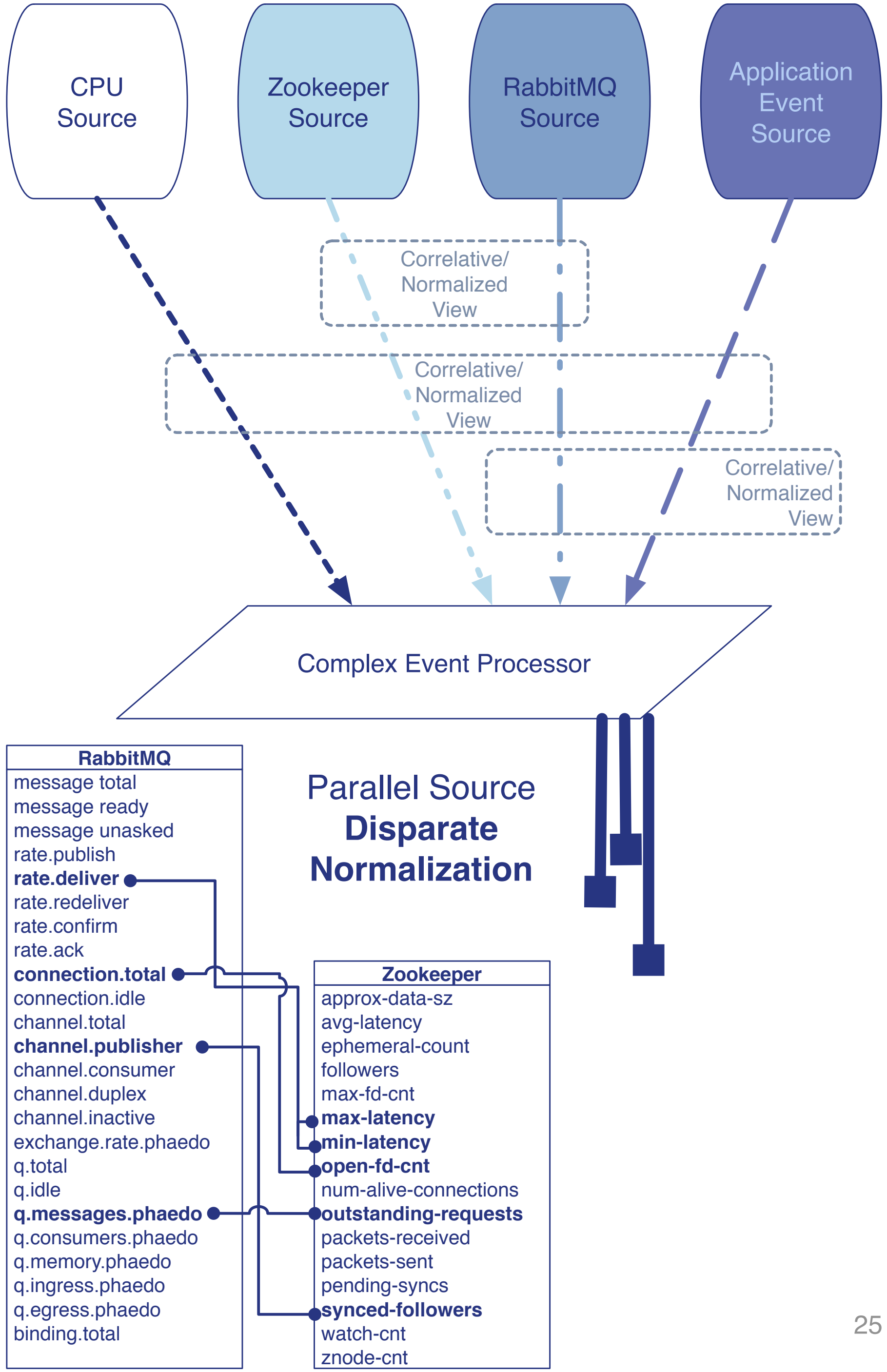
Affinity + Simple Case

Stream + Augmented Datasource

Parallel Stream

Frequency-Shifted Stream

- “Correlative/Normalized View”: Similar to a SQL “join” concept, we relate data fields in disparate stream sources
- Requires frequency mapping (sliding windows, state management, etc.)
- **Use Case Example: Messaging System and Zookeeper filesystem relationships**
  - vector time (event/observation based)
  - incoming/outgoing pipeline relationships
  - actor mapping
  - filesystem/messaging performance



Compound Approaches

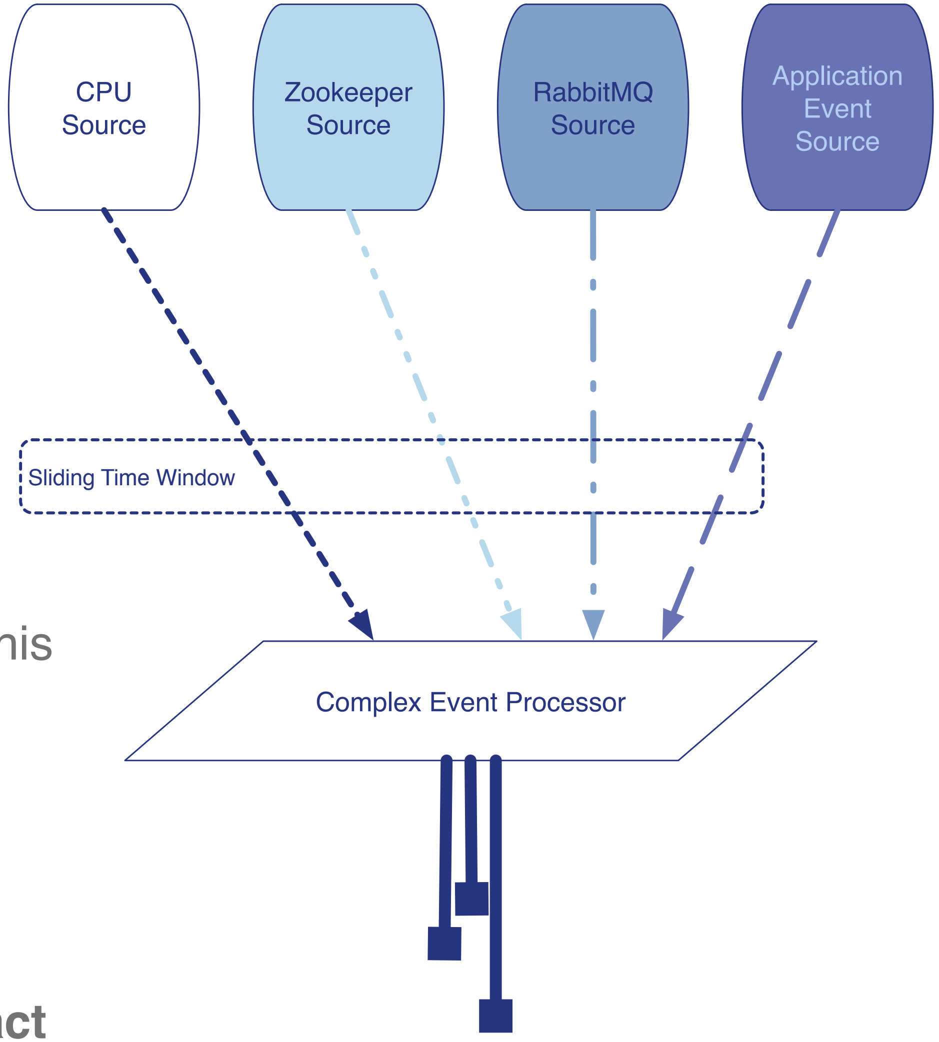
Affinity + Simple Case

Stream + Augmented Datasource

Parallel Stream

Frequency-Shifted Stream

- Not a simple problem, and is usually where the *"it's easier to just do this in situ"* argument comes up.
- Most sources do not publish at the same interval. To handle this we need a variety of techniques (some examples):
  - sliding time windows
  - state management (value looping)
  - relevancy-offset clocks (determined by "master events")
  - store and forward
- **Use Case Example: Application Environmental CPU Impact**
  - CPU published on time interval, leverage value looping
  - Application is event-driven, it's the master.



Parallel Source
Disparate Frequency

CPU	Zookeeper	RabbitMQ	Application
event_duration_ms event_timestamp_orig observed_timestamp observation_latency	event_duration_ms event_timestamp_orig observed_timestamp observation_latency	event_duration_ms event_timestamp_orig observed_timestamp observation_latency	event_duration_ms event_timestamp_orig observed_timestamp observation_latency

What does it take to get from design to best-practice?

If we take away nothing else…

Theory into Practice

Theory into Practice

In-situ is easy, but it’s **not going to work long term** — we need to gain real insight faster — as things happen.

@phaedo

Theory into Practice

In-situ is easy, but it’s **not going to work long term** — we need to gain real insight faster — as things happen.

Push analytics to the edge. We will see near-field analytics, edge-analytics, related-entity analytics, etc. **When you can’t push it to the edge, push it to the edge anyway.**

Theory into Practice

In-situ is easy, but it's **not going to work long term** — we need to gain real insight faster — as things happen.

Push analytics to the edge. We will see near-field analytics, edge-analytics, related-entity analytics, etc. **When you can't push it to the edge, push it to the edge anyway.**

In-transit analysis requires a **second-order approach to information and insight**, and requires we **divorce publisher and consumer**.

Theory into Practice

In-situ is easy, but it's **not going to work long term** — we need to gain real insight faster — as things happen.

Push analytics to the edge. We will see near-field analytics, edge-analytics, related-entity analytics, etc. **When you can't push it to the edge, push it to the edge anyway.**

In-transit analysis requires a **second-order approach to information and insight**, and requires we **divorce publisher and consumer**.

Messaging middleware is already mature for most of these design patterns, but hasn't been leveraged well for Big or scientific data yet. But it's the only mature technology looking at **data movement as a conceptual problem independent of content.**

Theory into Practice

In-situ is easy, but it's **not going to work long term** — we need to gain real insight faster — as things happen.

Push analytics to the edge. We will see near-field analytics, edge-analytics, related-entity analytics, etc. **When you can't push it to the edge, push it to the edge anyway.**

In-transit analysis requires a **second-order approach to information and insight**, and requires we **divorce publisher and consumer**.

Messaging middleware is already mature for most of these design patterns, but hasn't been leveraged well for Big or scientific data yet. But it's the only mature technology looking at **data movement as a conceptual problem independent of content.**

Never underestimate the power and placement of small computers. My watch is more capable than laptops only 2 generations ago. The age of **general compute is giving way to generally specialized** computers. They will make a huge difference to streaming larger and more complex data. We really can look at everything.