MetricHub

Why Metrics?

- Continuous improvement
  - Monitor KPIs
  - Process improvement- don't guess; decide empirically
- Capacity management
  - Manage demand
  - Chase demand
- Anomaly detection
  - Detect events
  - Respond in real time
  - Crisis diagnosis

Motivation 1: The value that a terminal adds is in the context of the overall supply chain.

So metrics should not just be internally focused and reported

Motivation 2: Variability is costly (Operations management 101)

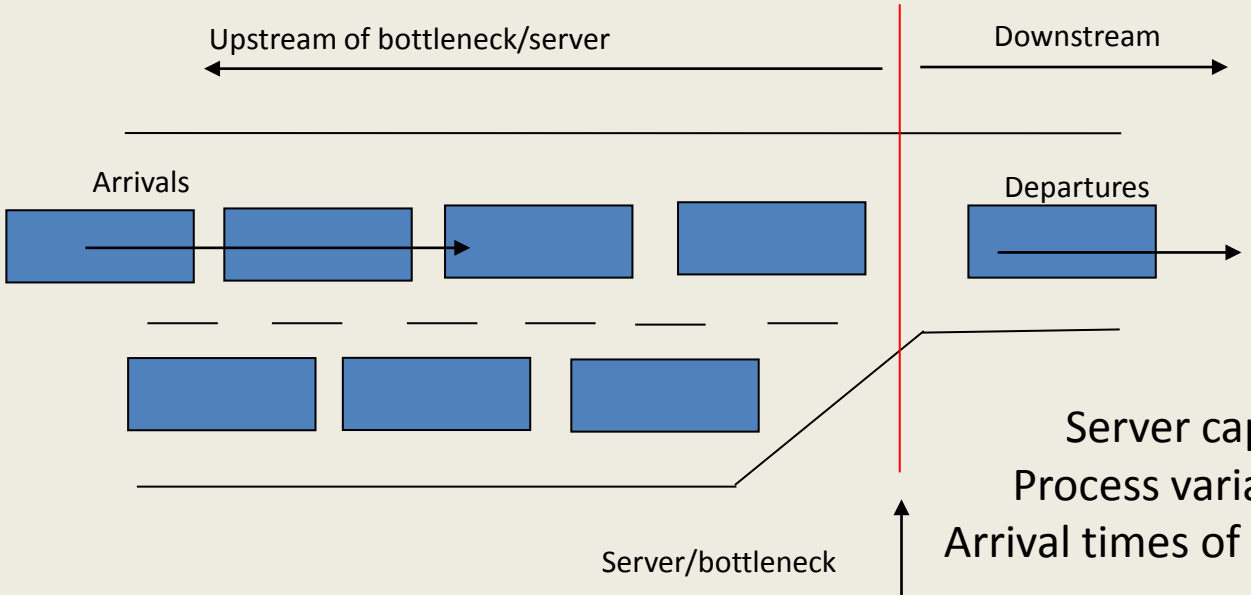
….But Variability is imposed by external actors

Arrival patterns determined by:

Other terminals/warehouses
Haulers
Traffic conditions
…. Ect

Information typically lies outside your system

Departure patterns determined by:



Server capacity
Process variability
Arrival times of trains

Architecture

MetricHub: architecture

Problem: Integration

- Lots of systems:
  - Different protocols
  - Different data formats
- You don’t control the system:
  - What if they won’t change their system
  - What if they change their system !!

The usual solutions to this problem?

A) Large Bespoke integration project
B) Get everyone to adopt a Standard
C) Message translation/Semantic mediation

For all these solutions you need to
…. Keep updating system whenever any system changes

But we just want to calculate metrics

Q **Translating messages is difficult & tedious**
How can clients or services function properly when some of the content in the messages or media types they receive is **unknown** or **when the data structures vary**?

Surely there must be a simple way?

I **We Know what metrics we want to calculate** These metrics only require a very small part of the data being transmitted.

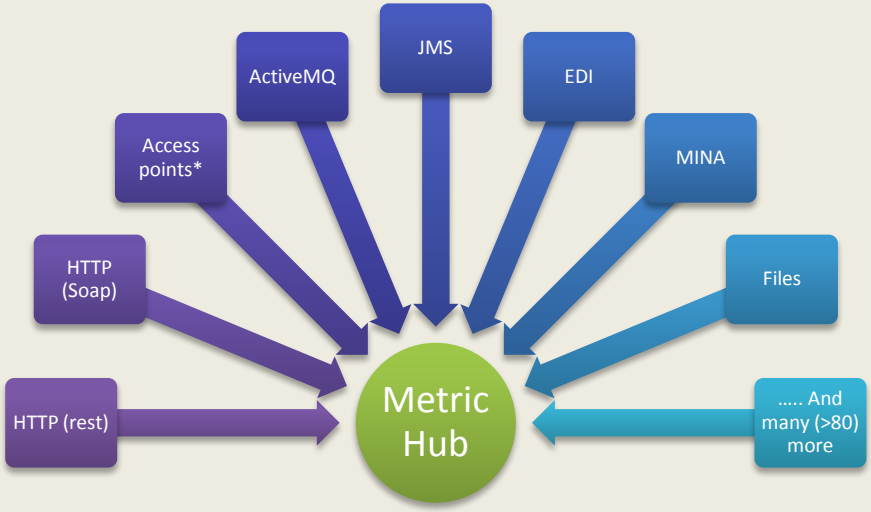
A Design the client or service to **extract only what is needed**, ignore unknown content, and **expect variant data structures**.

How?

Avoid "reinventing the wheel"

Concise
Application
Messaging
Exchange
Language

Accept more than 80 protocols → Accept more that 20 formats → Convert to arbitrary JSON structure

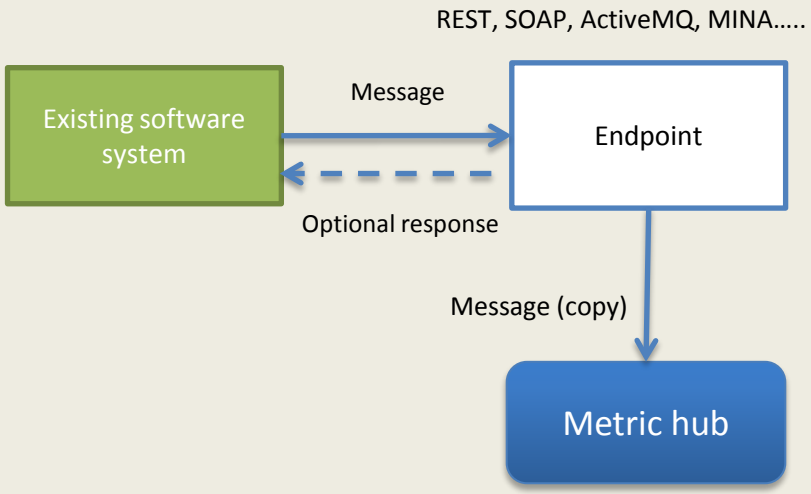


bindy	protobuf
castor	serialization
csv	soap
crypto	syslog
dozer	tidy markup
flatpack	xml beans
gzip	xml security
hl7	xstream
jaxb	zip

*With Lime client developed elsewhere

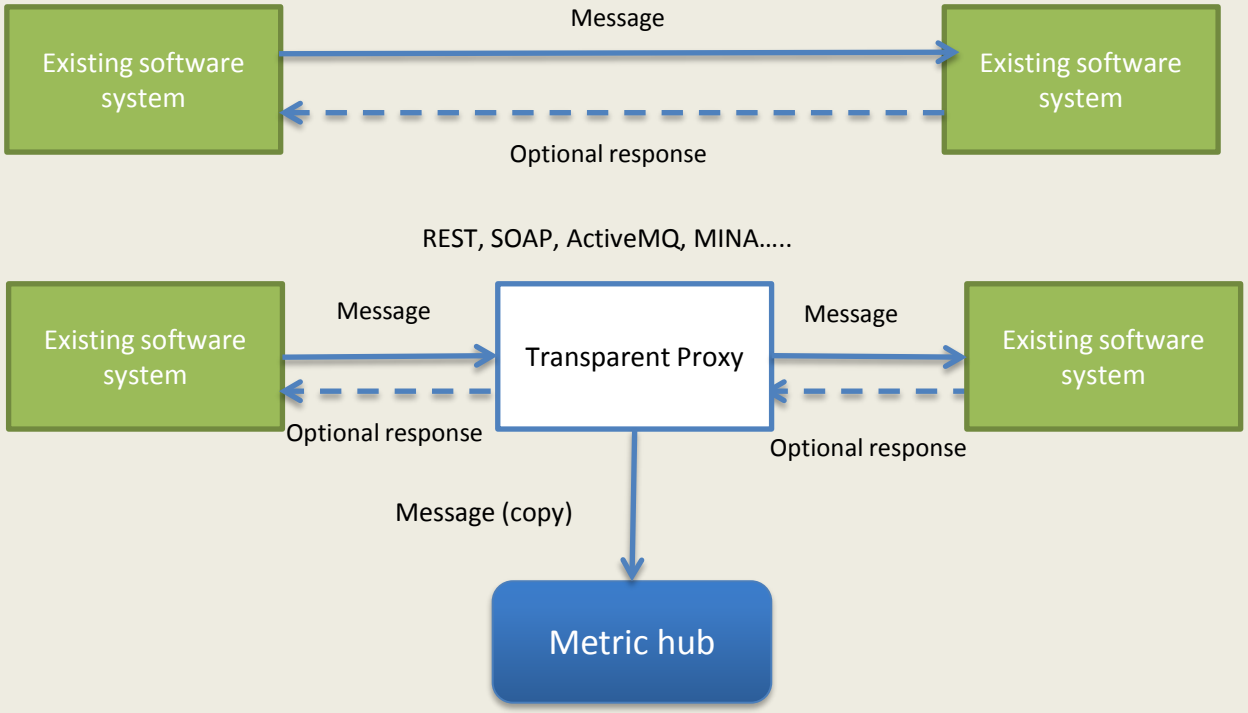
*With smooks plugin: http://www.smooks.org

So we have a tolerant reader

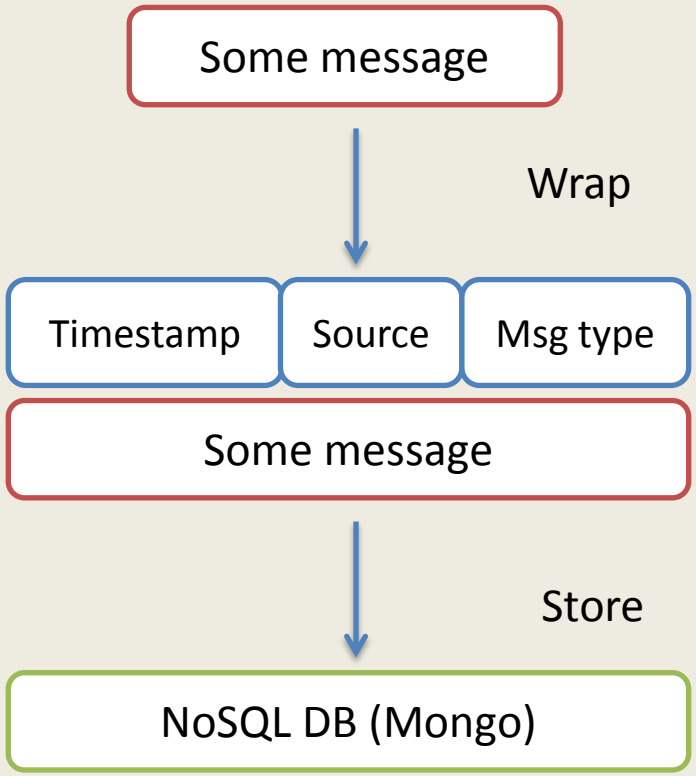


Q But what if you can't modify the code of the system

A Interception mode



A Storing unknown arbitrary structure



Can store arbitrary data.

No need to specify schema

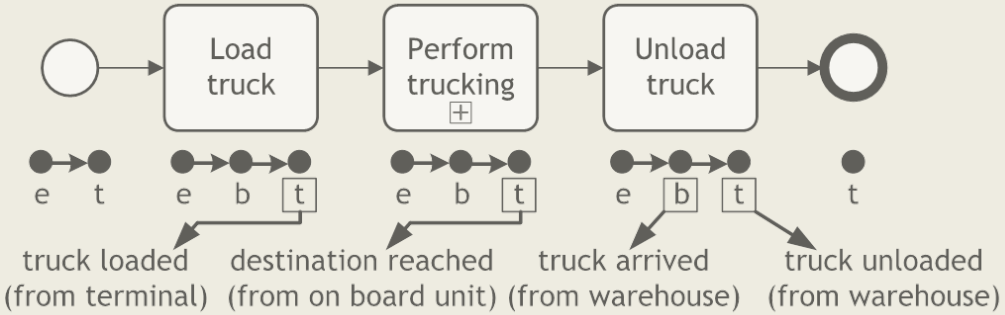
So don't need to know message schema

Arbitrary hierarchical document

```
{
  /* Sender identifier e.g. you are the ecohubsLastmileApp */
  "sender"  : String,
  /* Time stamp
   * UTC ISO 8601 date format e.g. "2014-05-22T08:42:55.425Z"
   */
  "eventTime"  : String.validDate(DateFormat.ISO8601),
  /* Type of event e.g. "slot_allocated, truck_arrived,
   * container_loaded", "container_delivered" */
    "type"     : String,
  /* Data field contains arbitrary json
   * Events of the same "type" should have the same data schema */
   "data" : JSONObject
}
```

So now you have data in a database with no schema.

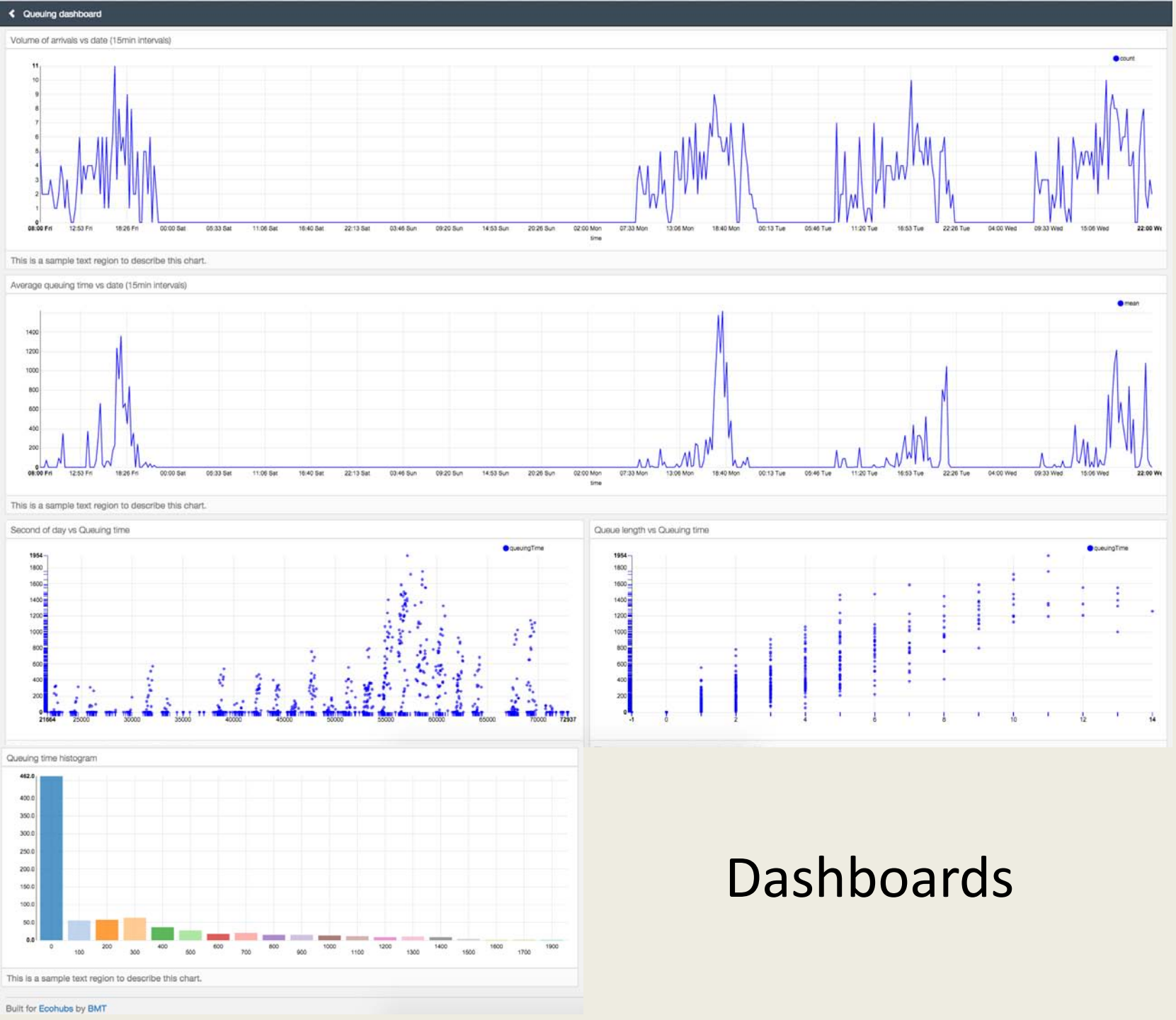
Q How do you calculate useful metrics?



i Most metrics are associated with events.

A So detect events

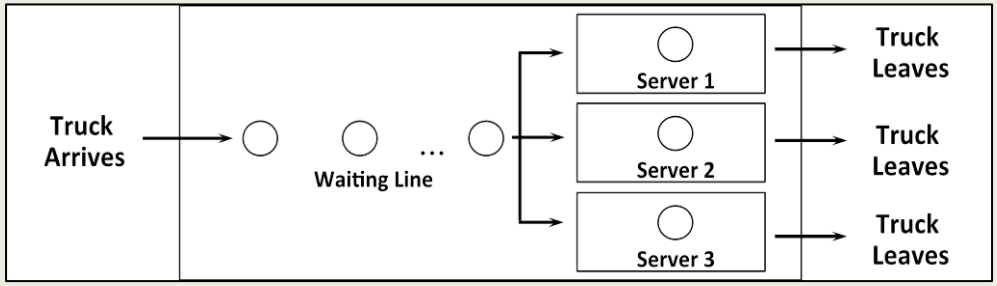
```
insert into TripBetweenTerminals(truckID, t)
        select A.truckID,  B.timestamp - A.timestamp
        from pattern [
        every-distinct(A.truckID, 30 min)
                A=LeaveOtherTerminal ->
                B=ArriveMyTerminal(truckID=A.truckID)
        ]
```


Queuing dashboard
Volume of arrivals vs date (15min intervals)
count
time
This is a sample text region to describe this chart.
Average queuing time vs date (15min intervals)
mean
time
This is a sample text region to describe this chart.
Second of day vs Queuing time
queuingTime
Queue length vs Queuing time
queuingTime
Queuing time histogram
This is a sample text region to describe this chart.
Built for Ecohubs by BMT

Dashboards

Trucks arrive following historic arrival pattern. Information from TAS used to identify when they enter terminal queue

The trucks join a queue for "check in" they provide details (number plate, trucking company, delivery reference, ISO code, dangerous goods information) and receives an info sheet. This process takes some time



The trucks join a queue for "check in" they provide details (number plate, trucking company, delivery reference, ISO code, dangerous goods information) and receives an info sheet. This process takes some time

Interesting stuff

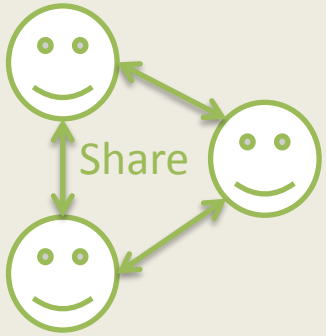
Queue length
Queuing time
Temporal patterns

Punctuality
Caused by:
- Haulier Company variation
- Other locations (warehouses terminals)
- Traffic

Over/under capacity.
Differences in speed varies with shifts

MetricHub

Real time data collection &
metric **sharing** Between external
actors



Consume data from any source

Easy sharing to anyone

Flexible metrics

Reactive event detection