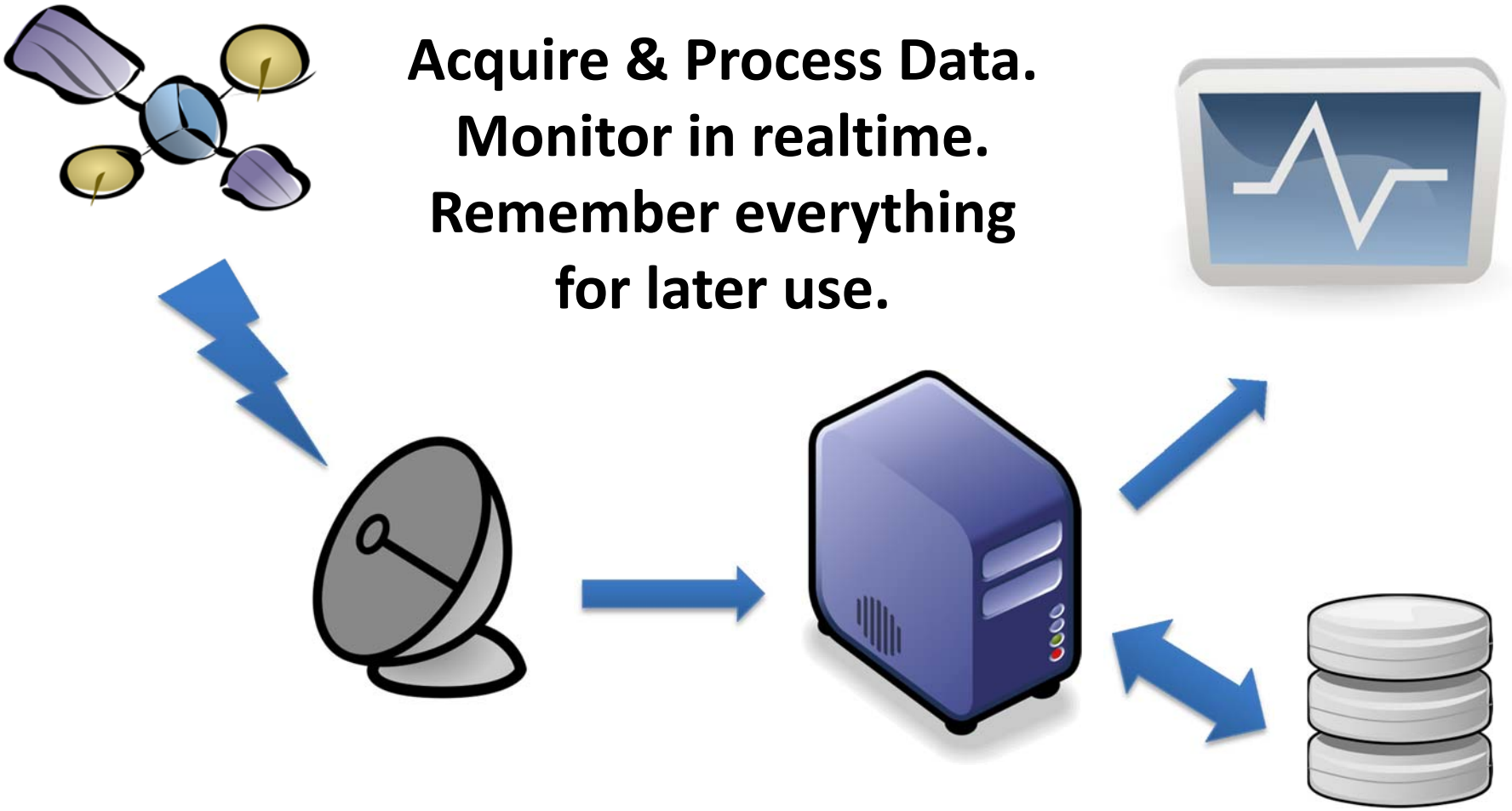
High performance Timeseries storage in ISIS

Nicolas Champsavoir

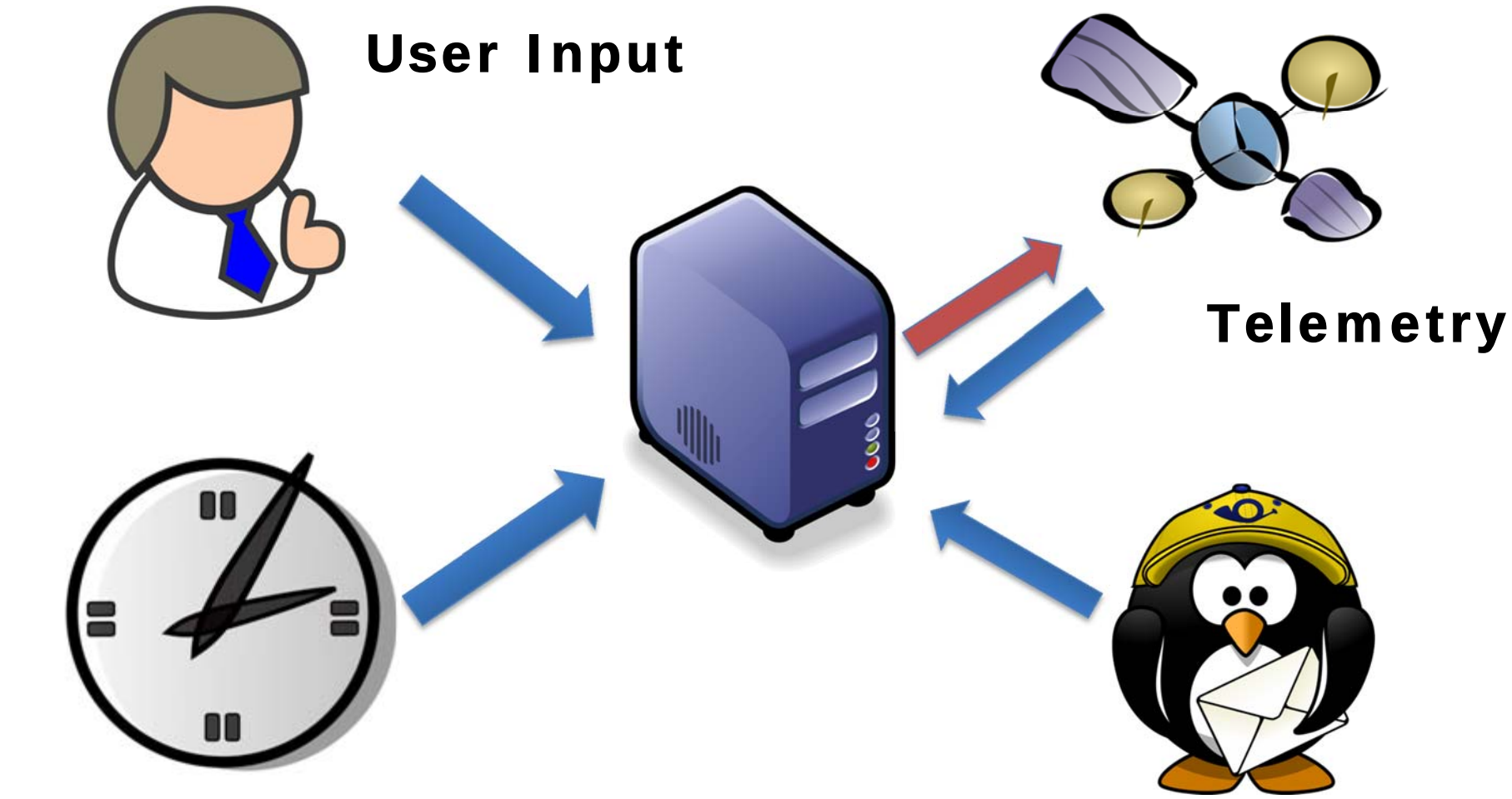
© 2015 CNES. Published by The Aerospace Corporation with permission.

- **A CNES Initiative**
- **A Set of Standards**
- **A Mission Control Center Product Line**

Typical Control Center

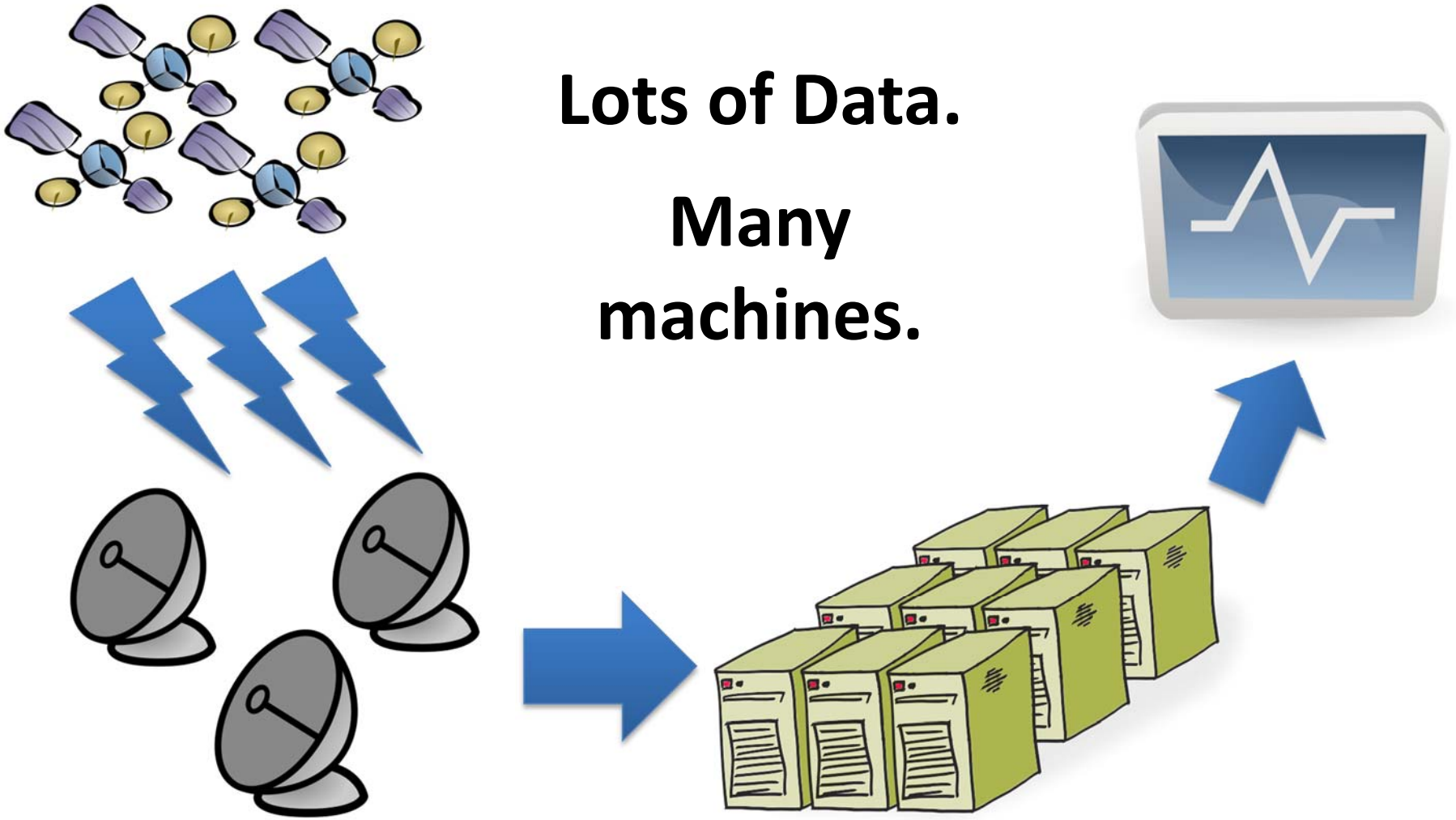


MCS is Event-Driven



Typical Control Center in 2030?

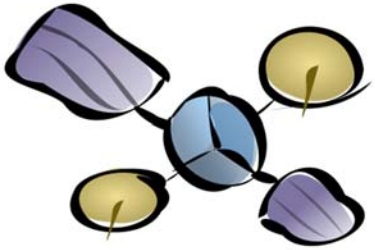
Or rather…



Lots of Data.

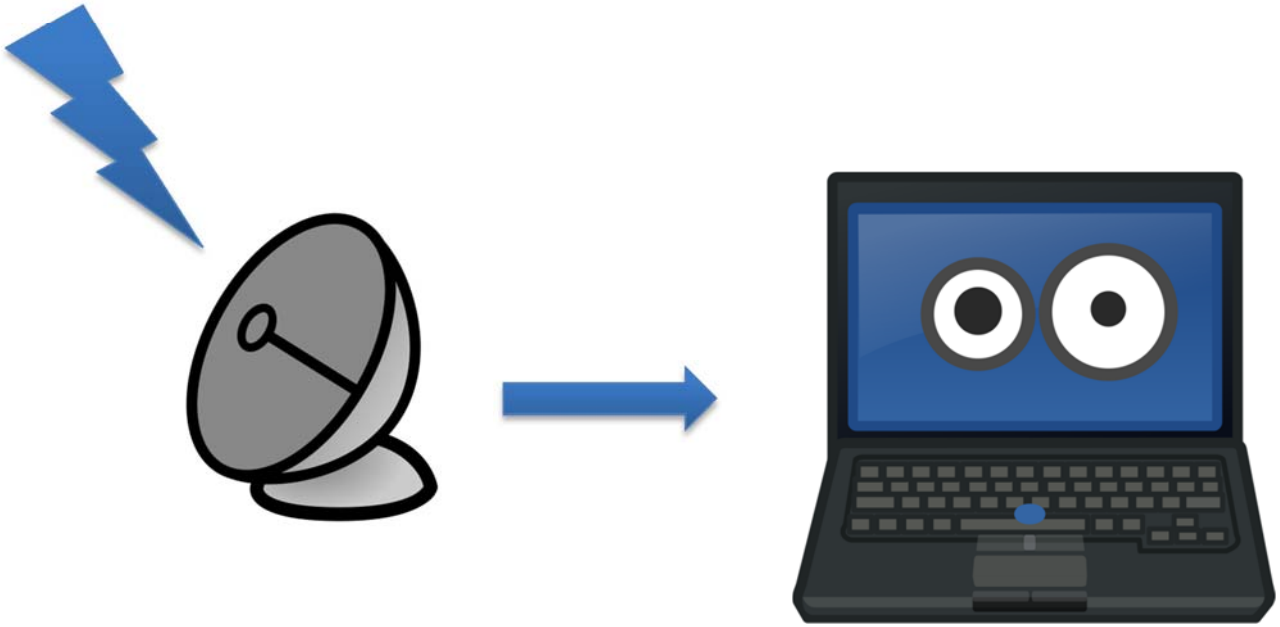
Many machines.

And also…



Reasonable Amout of Data.

On my laptop.

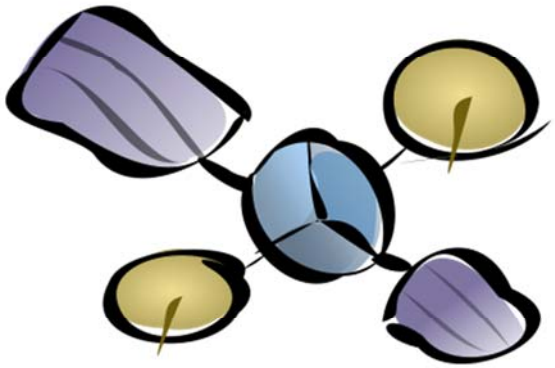


All the data. All the time.

In any case, performance matters.

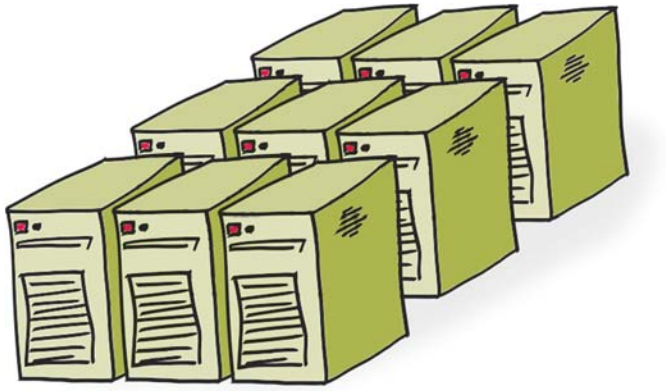
How to design a storage system for space data that scales ?

Requirements...



Timeseries are first class citizens

Lightweight and easy to install



Scalable

Consultative Committee
CCSDS
for Space Data Systems

asi
agenzia spaziale italiana
CSA ASC
cnes
CNSA
DLR
esa
INPE
JAXA
NASA
UK SPACE
AGENCY

CCSDS Mission Operations

Interoperability at the Application Level
Technology & Transport Agnostic

Monitoring & Control Services Specifications

And more to come…

A Toolkit to define new Service Specifications

Message Abstraction Layer

LIST SONGS, PLAY, STOP, PAUSE…

Message Abstraction Layer

Message Abstraction Layer

Defines the common vocabulary to exchange messages

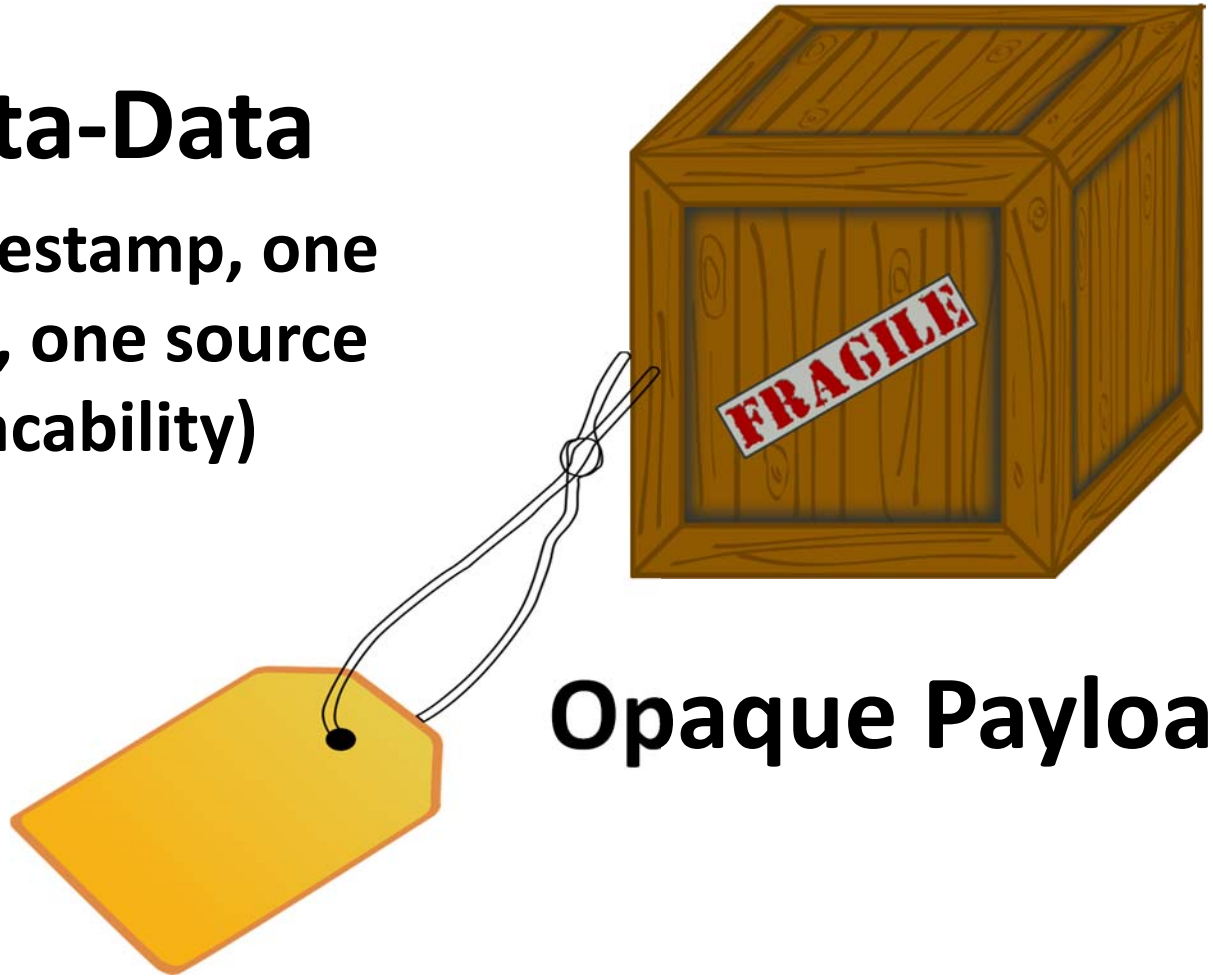
SEND, RECEIVE, SUBSCRIBE, UNSUBSCRIBE…

Not the technology.

Common Object Model

Meta-Data

ID, Timestamp, one parent, one source (tracability)



Opaque Payload

Common Object Model

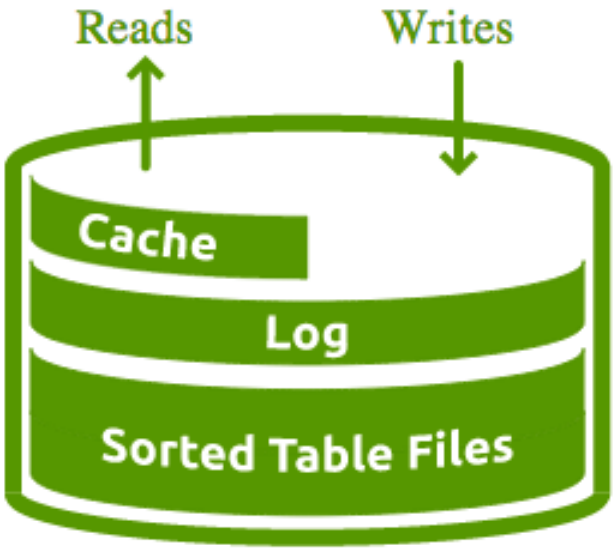
Defines the common vocabulary to identify and search for space data

WHAT, WHERE, WHO, WHEN…

Not the technology.

Introducing DataStore

***Fast**, Small, Open-source, dependency-free, embedded key/value data store*

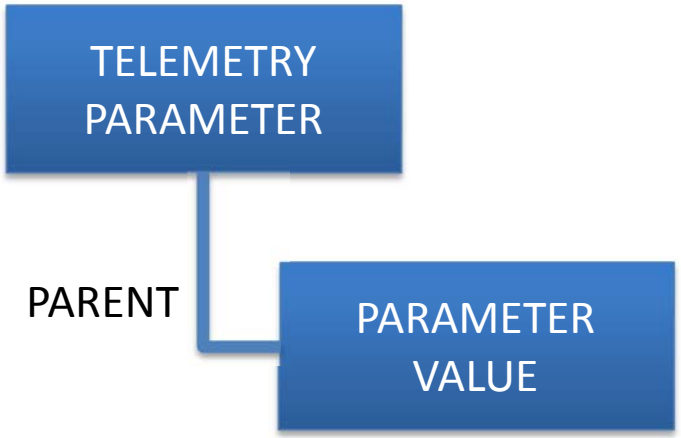


Arbitrary byte arrays
Sorted by keys
Compressed storage
Basic operations:

Get(), Put(), Del(), Batch()

Consultative Committee
CCSDS
for Space Data Systems

COMMON OBJECT MODEL



Find all parameter values for SATMODE between 2014-02-14 15:00 and 2014-04-13 02:00

Iterate on all keys between

SATMODE	2014-02-14 15:00	00000000

and

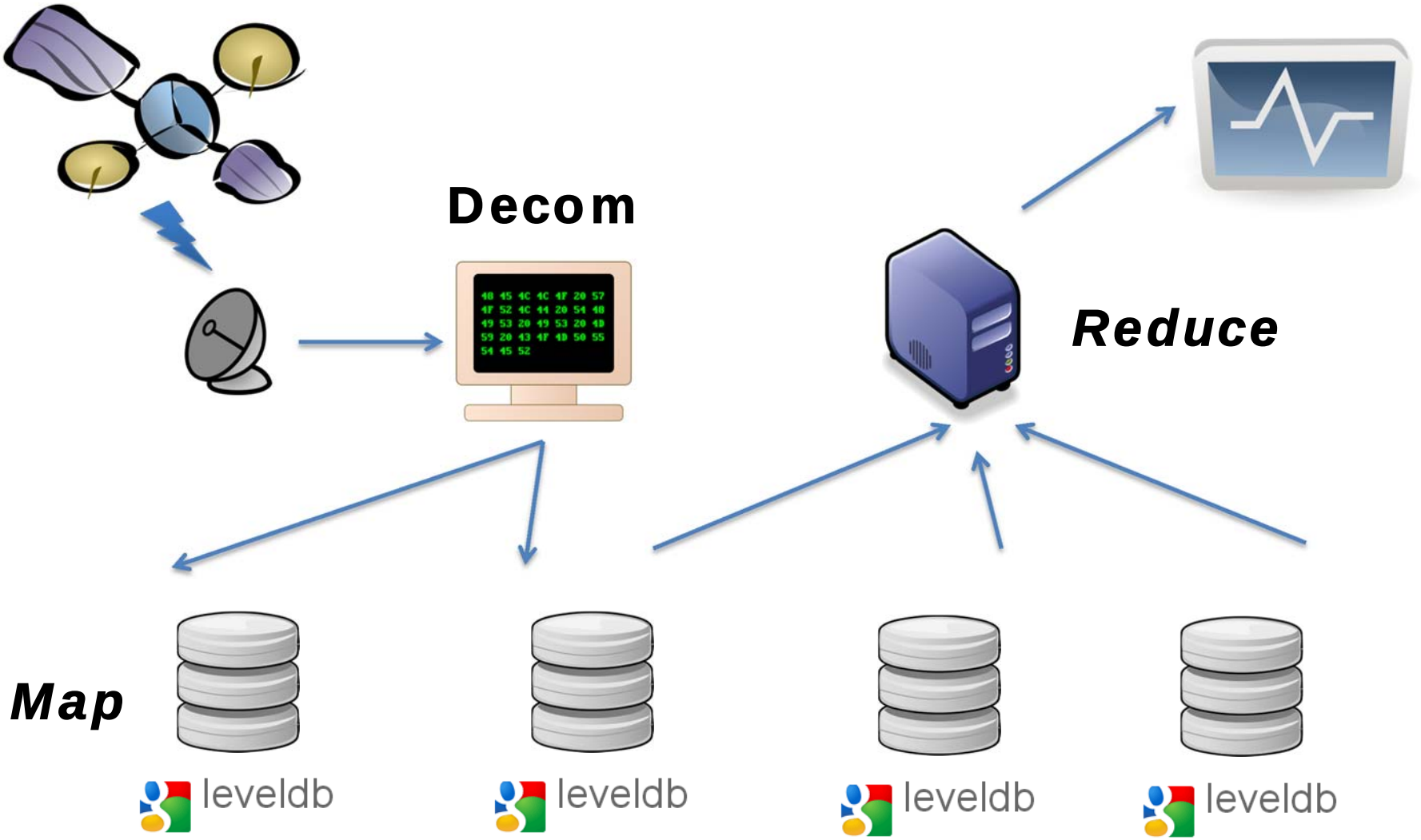
SATMODE	2014-04-13 02:00	00000000

From LevelDB GitHub page

LevelDB: version 1.1
Date: Sun May 1 12:11:26 2011
CPU: 4 x Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz
CPUCache: 4096 KB
Keys: 16 bytes each
Values: 100 bytes each (50 bytes after compression)
Entries: 1000000
Raw Size: 110.6 MB (estimated)
File Size: 62.9 MB (estimated)

readrandom : 60,000 ops/sec;
readseq : 0.476 micros/op; 2.1 million ops/sec

Telemetry / Events



Intelligent socket library for messaging

Many kinds of connection patterns
    REQ/REP, PUB/SUB, PUSH/PULL…

Multiplatform, multi-language (30+)

Fast (8M msg/sec, 30usec latency)

Small (20K lines of C++ code)

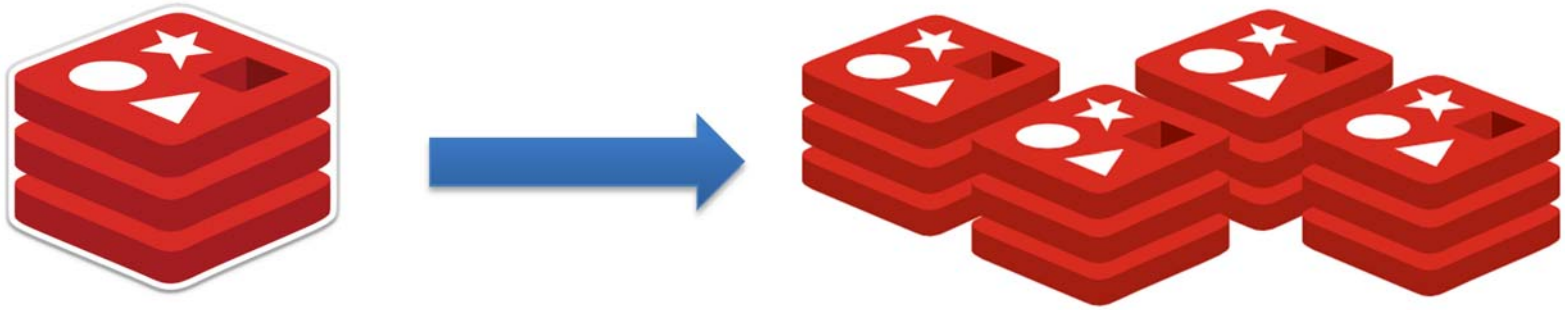
Open source LGPL (large community)

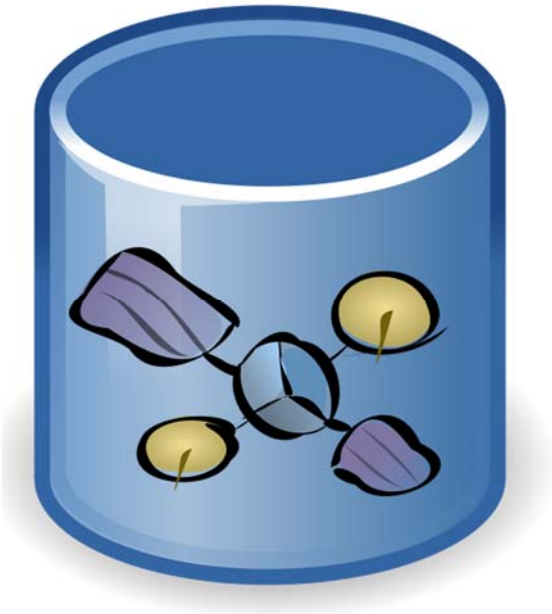
```
#
#   Hello World client in Python
#   Connects REQ socket to tcp://localhost:5555
#   Sends "Hello" to server, expects "World" back
#

import zmq

context = zmq.Context()

#  Socket to talk to server
print("Connecting to hello world server…")
socket = context.socket(zmq.REQ)
socket.connect("tcp://localhost:5555")

#  Do 10 requests, waiting each time for a response
for request in range(10):
    print("Sending request %s …" % request)
    socket.send(b"Hello")

    #  Get the reply.
    message = socket.recv()
    print("Received reply %s [ %s ]" % (request, message))
```

An open source in-memory **key-value cache** and **store**.

Contains strings, hashes, lists, sets, sorted sets…

Fast **Master / Slave replication** accross many machines.



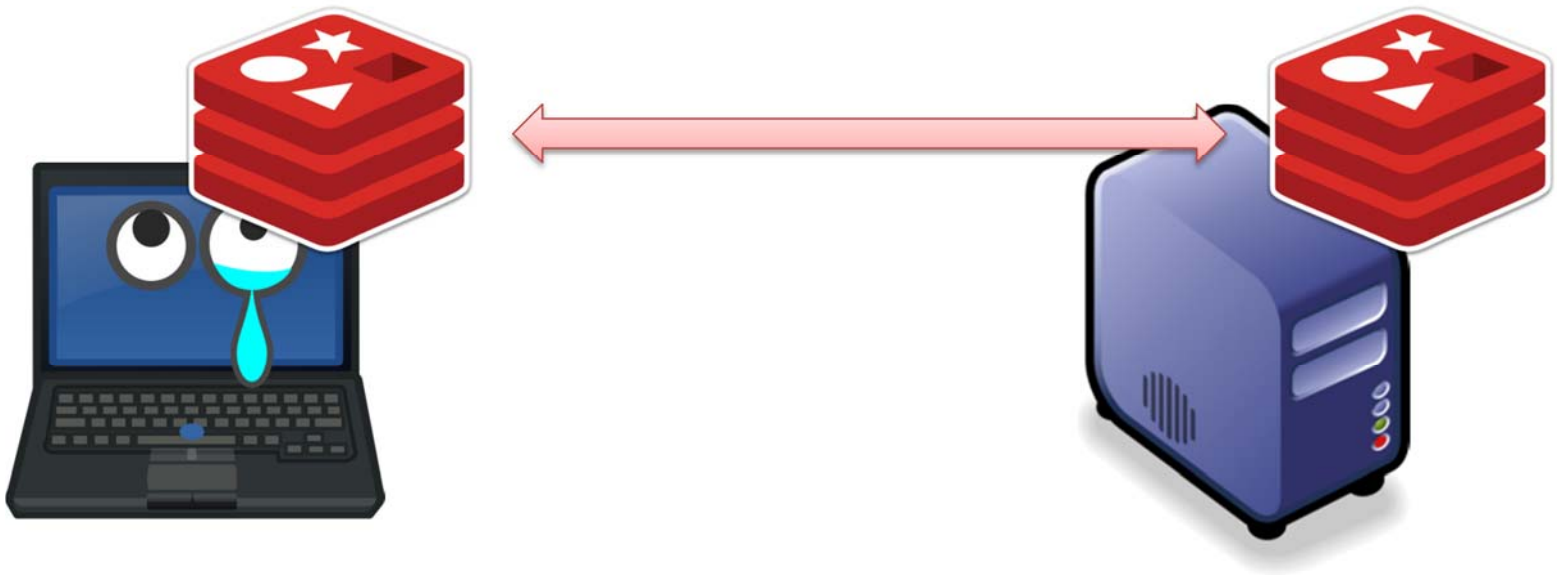


Mission Database

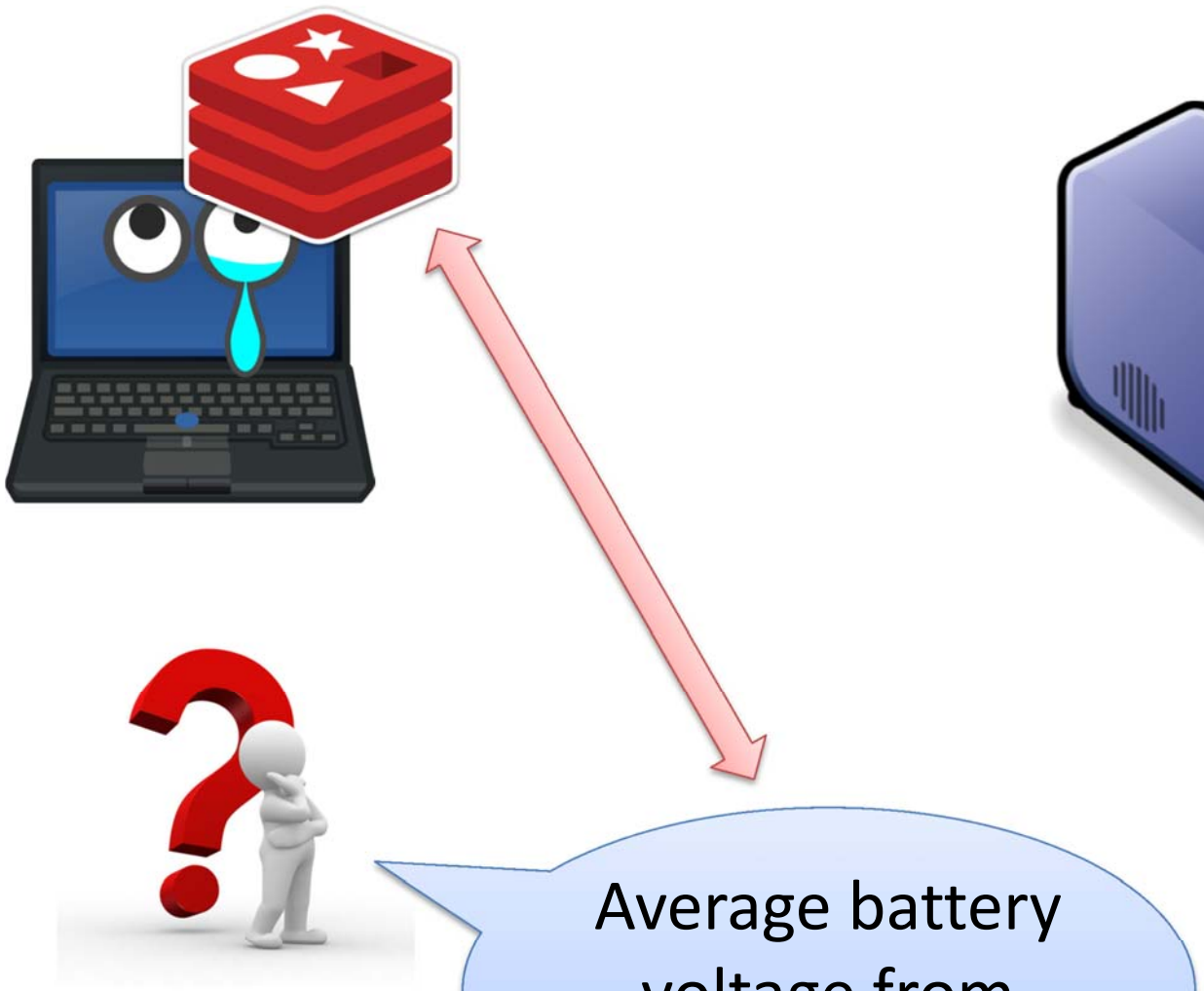
Ground Catalogs

{JSON}

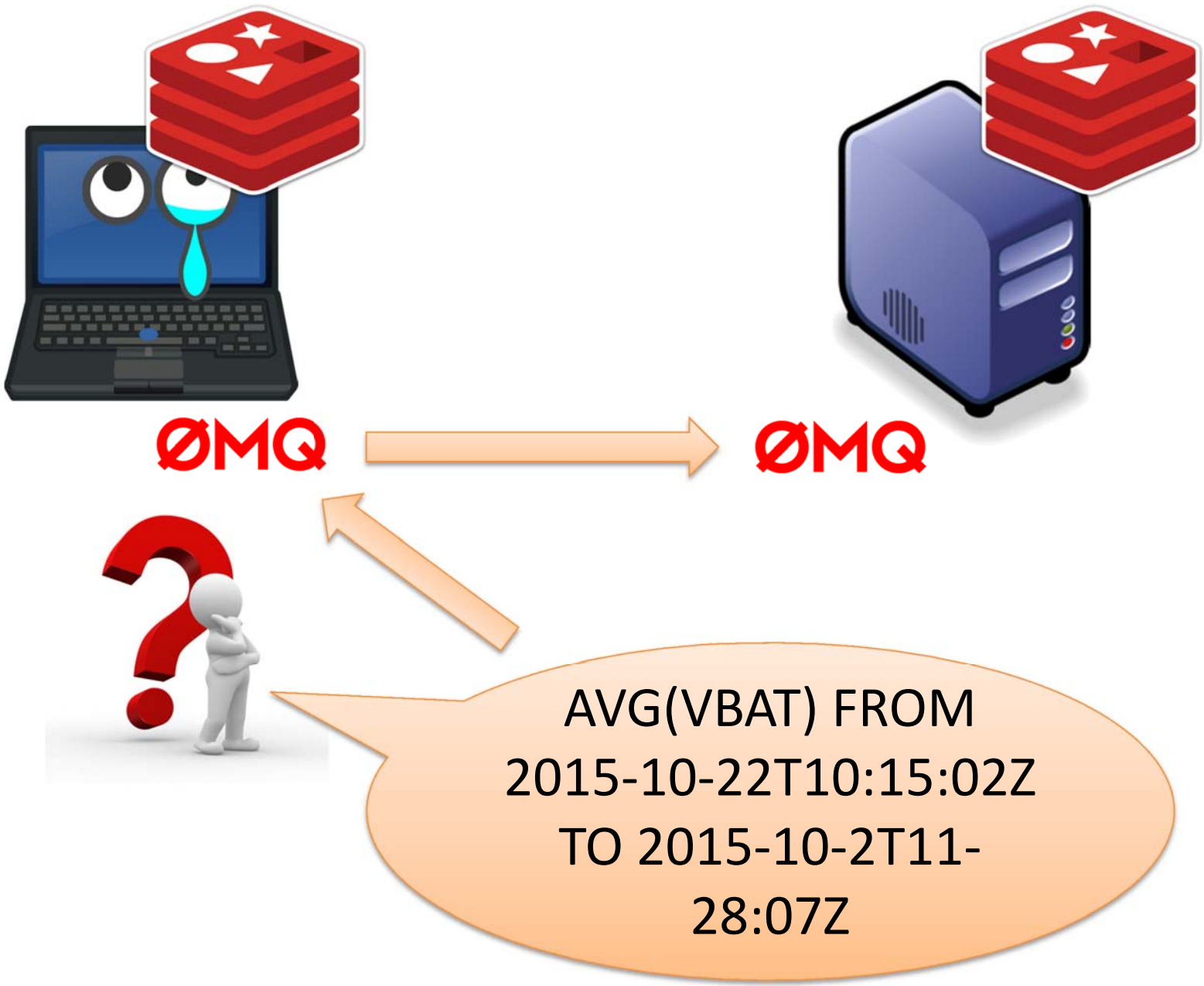
{JSON}

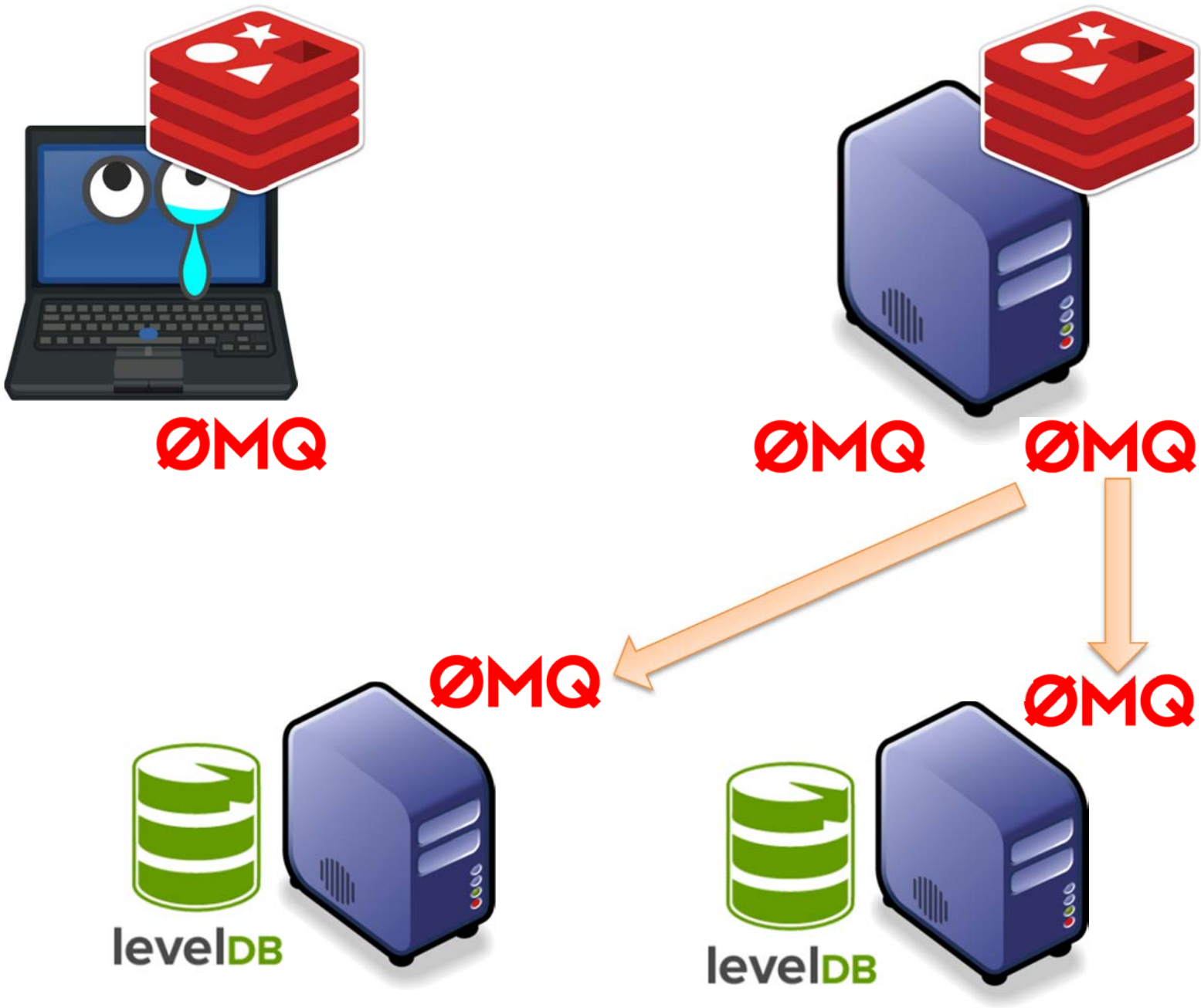


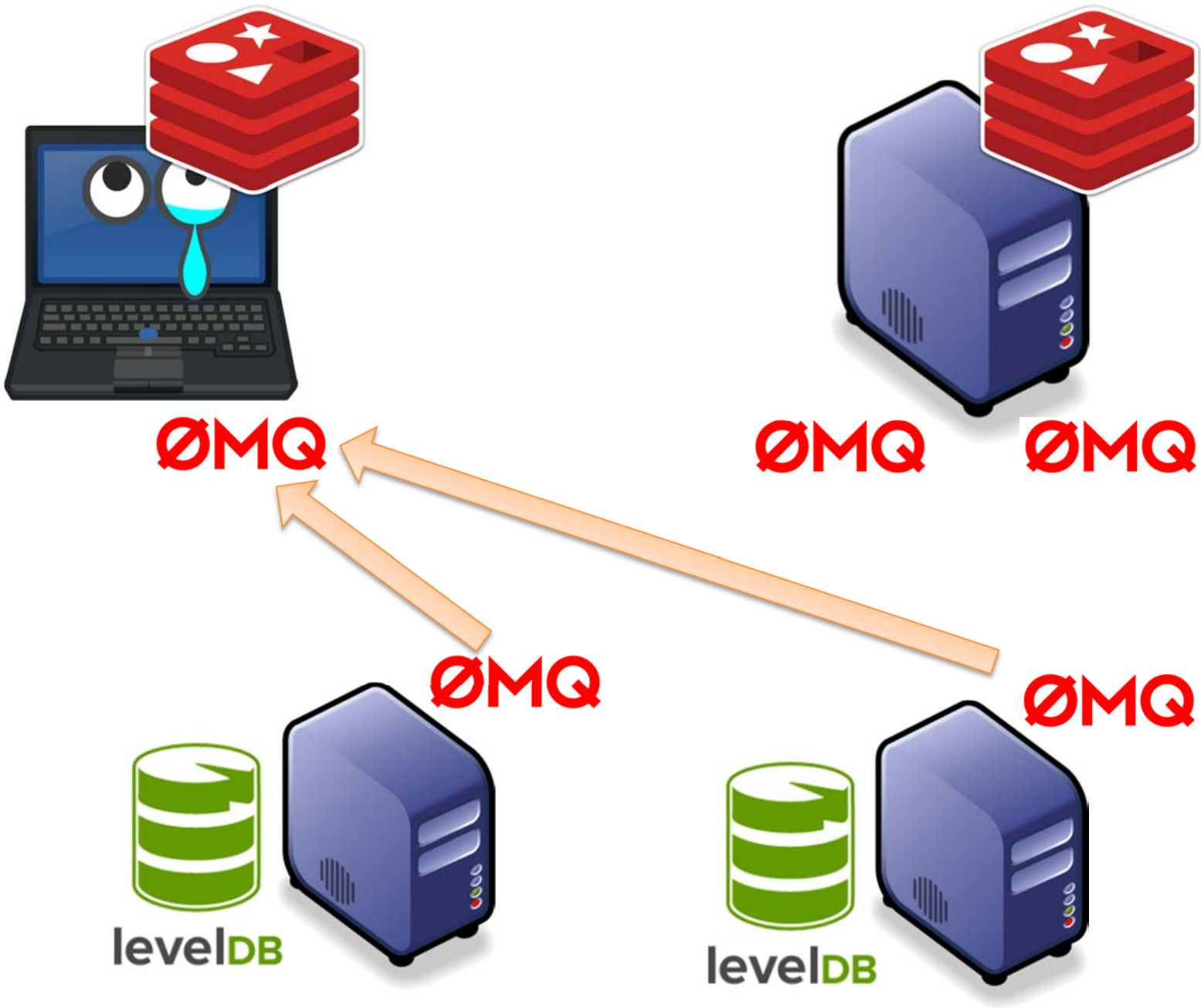
Average battery voltage from AOS to LOS ?


Average battery
voltage from
AOS to LOS ?

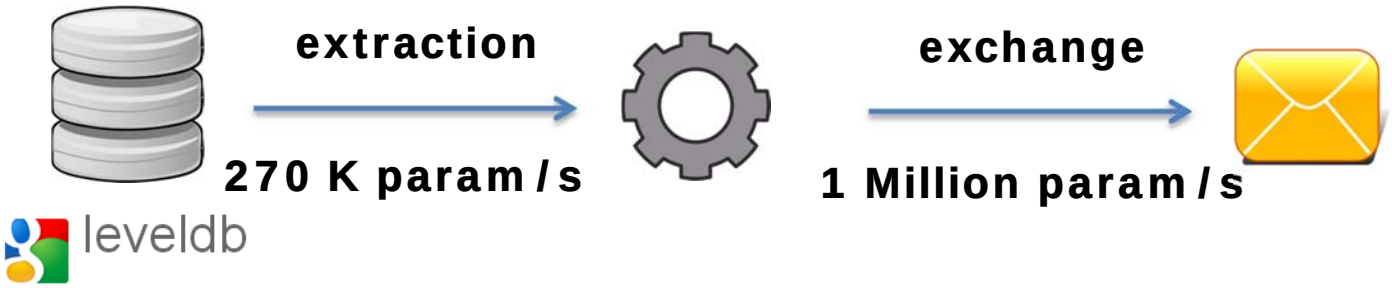
AVG(VBAT) FROM
2015-10-22T10:15:02Z
TO 2015-10-2T11-
28:07Z


ØMQ
ØMQ
AVG(VBAT) FROM
2015-10-22T10:15:02Z
TO 2015-10-2T11-
28:07Z

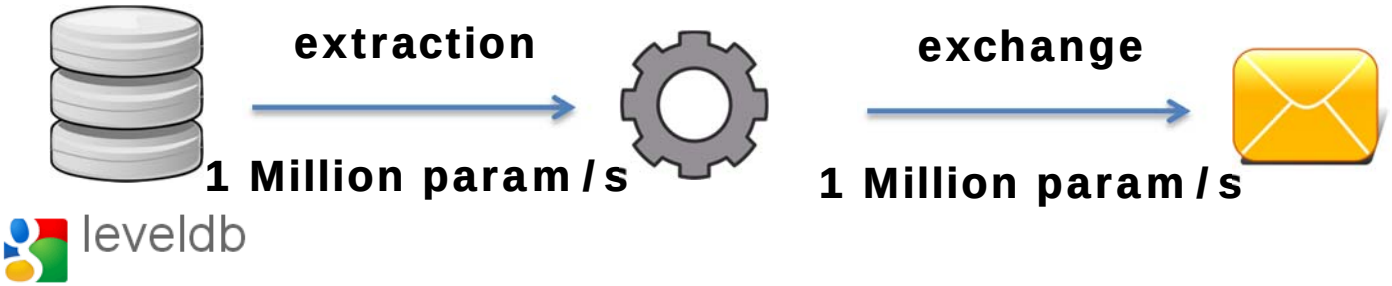

ØMQ
ØMQ
ØMQ
ØMQ
ØMQ
levelDB
levelDB


ØMQ
ØMQ
ØMQ
ØMQ
ØMQ
levelDB
levelDB

Performances



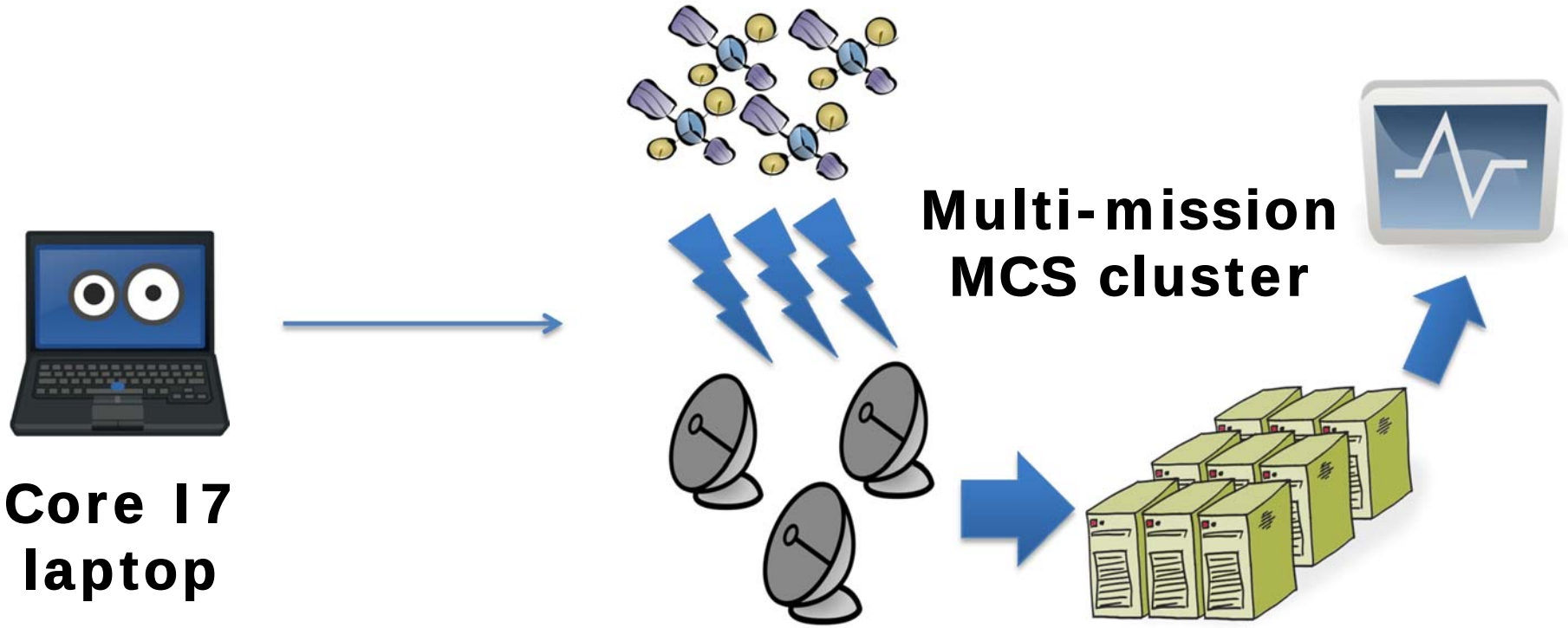
Prototype



ISIS Target

Core I7 laptop

Now just scale it



Questions ?