BBC...

CouchDB at the BBC

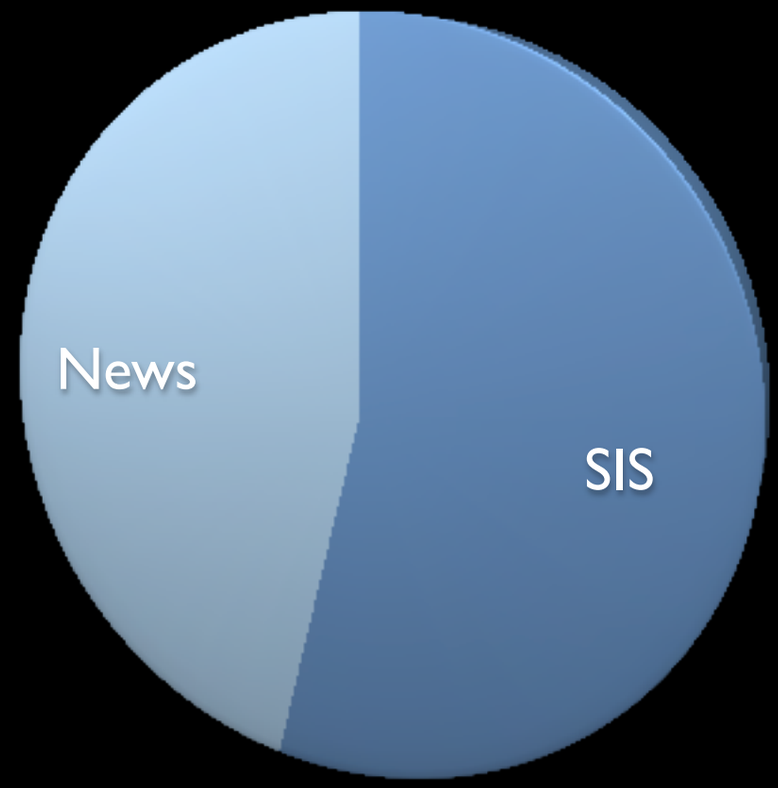
- In production
- Key Value Store - not a MapReduce, nor a document database*
- Operations more important than features
- A small - but important - feature in a much larger infrastructure

the platforms

- Until recently, two internet platforms

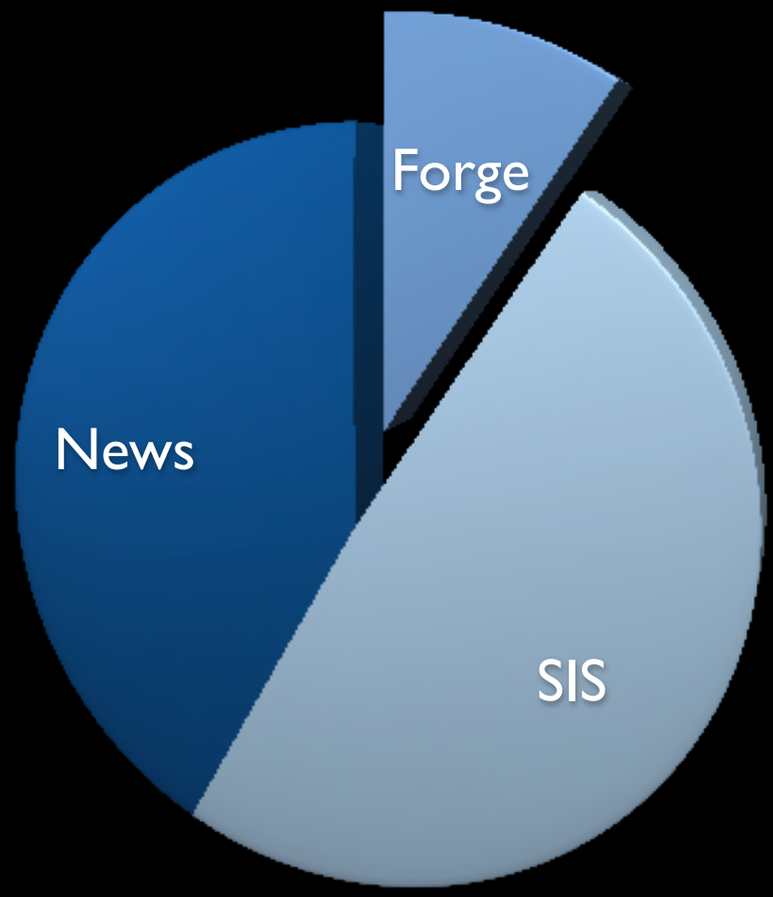
- news.bbc.co.uk
- www.bbc.co.uk

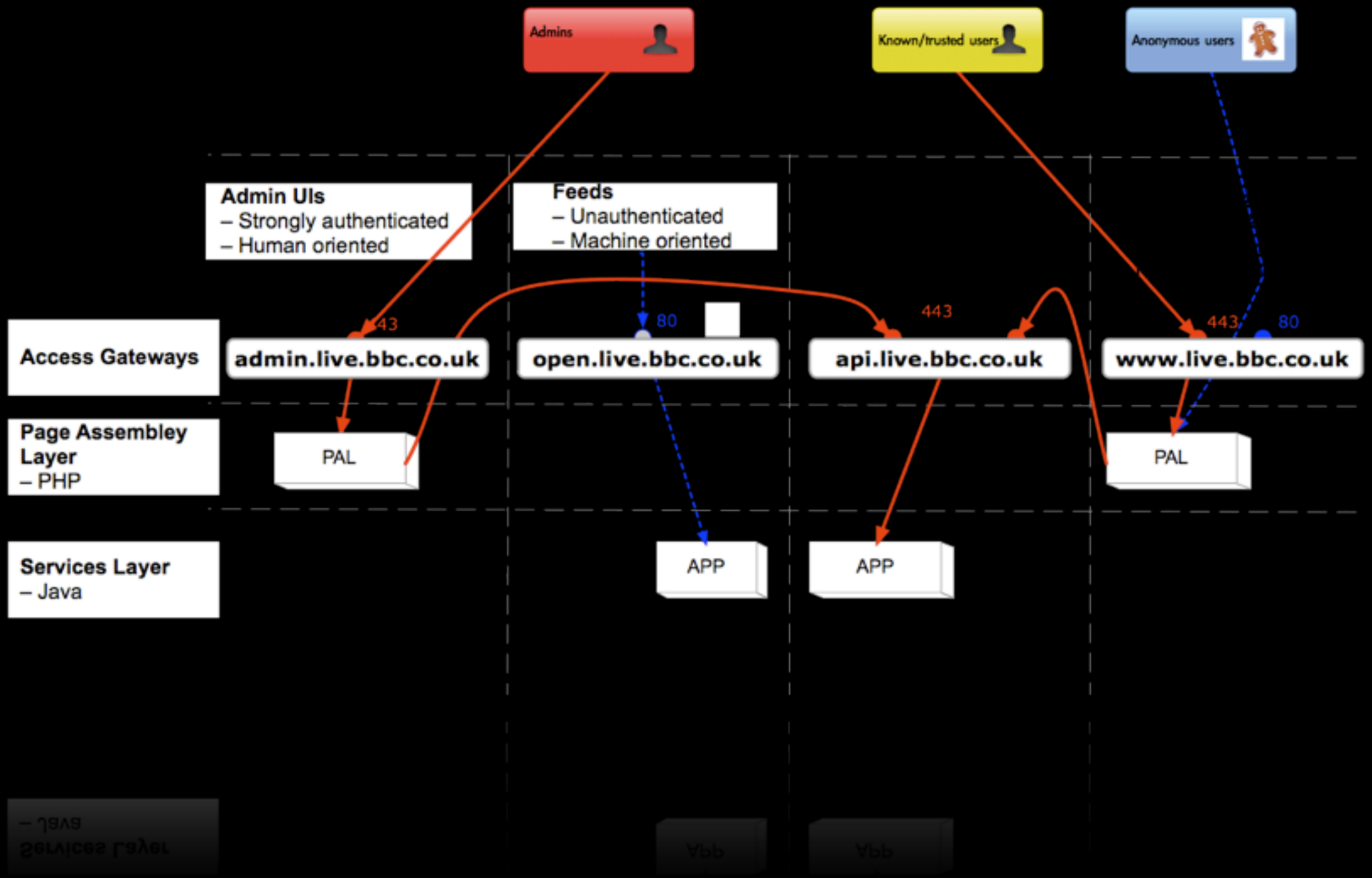
- Both are essentially static



the platforms

- There is now a third
- Forge
  - PHP - Zend
  - Lots of memcached
  - Java - Spring - Tomcat




Admins
Known/trusted users
Anonymous users
Admin UIs
– Strongly authenticated
– Human oriented
Feeds
– Unauthenticated
– Machine oriented
Access Gateways
443
80
443
443
80
admin.live.bbc.co.uk
open.live.bbc.co.uk
api.live.bbc.co.uk
www.live.bbc.co.uk
Page Assembley Layer
– PHP
PAL
PAL
Services Layer
– Java
APP
APP

Forge (well, ...)

Forge needs a key-value store

- Not everything needs to be ACID
- Replication of MySQL is not easy
- Scaling of MySQL isn't too easy either
- ... but we do have MySQL for those times when it's still appropriate

- NoSQL

Our problems

- Some of the most important considerations ...

Our problems: Don't know what we're doing

http://www.flickr.com/photos/plunkmasterknows/357836855

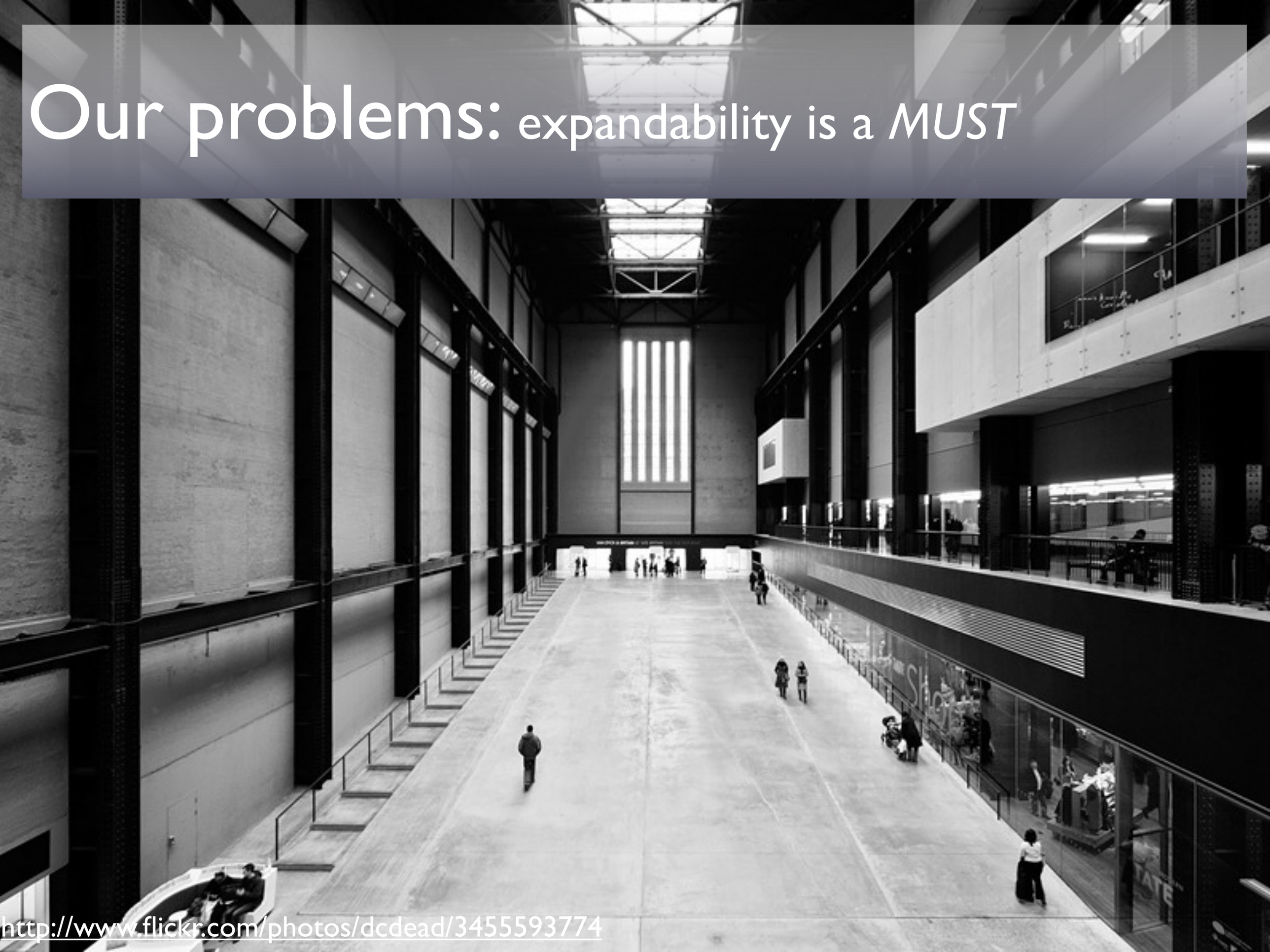
Our problems: Don't know how we'll be used

http://www.flickr.com/photos/48116183@N00/536340489/

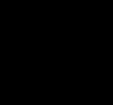
Our problems: Can't have *EVERYONE* trying to figure this out

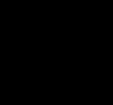
http://www.flickr.com/photos/not_words_but_in_things/356632727

Our problems: expandability is a *MUST*

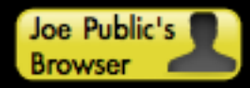


http://www.flickr.com/photos/dcdead/3455593774

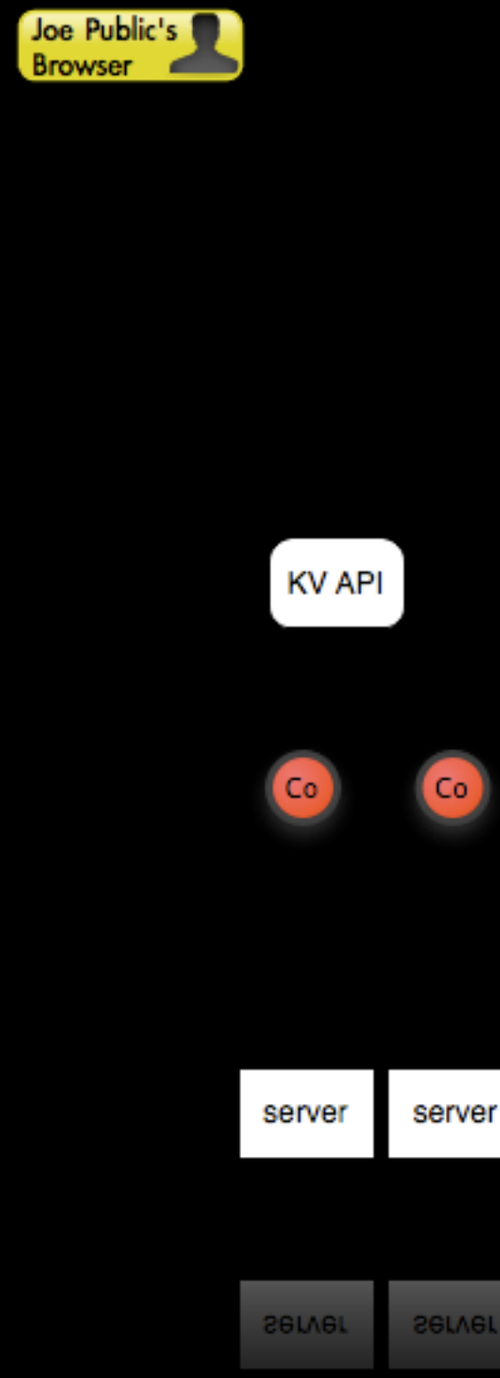




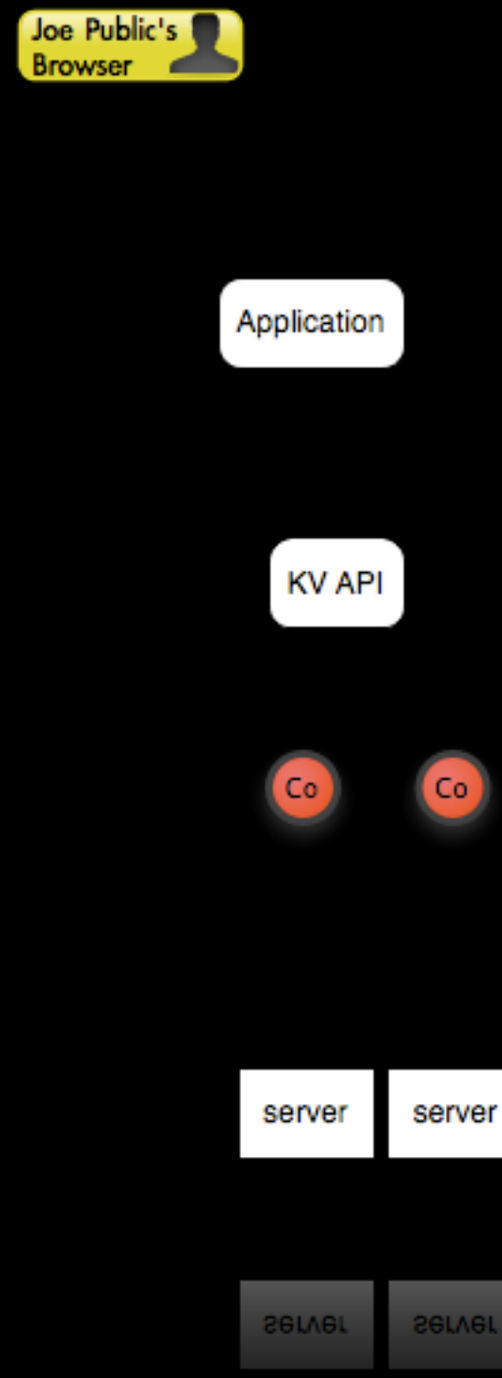
A typical setup?



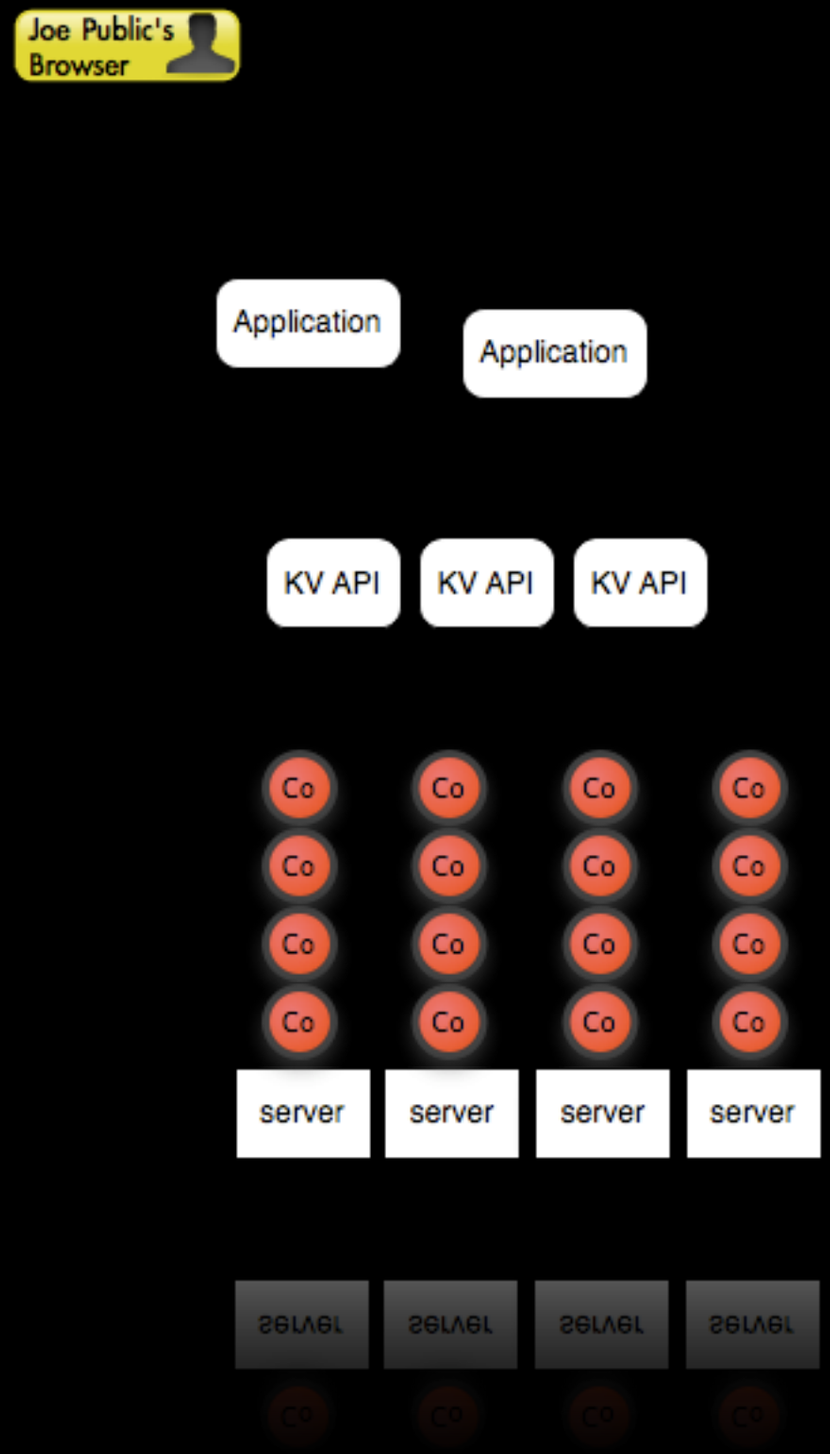
Large data sets



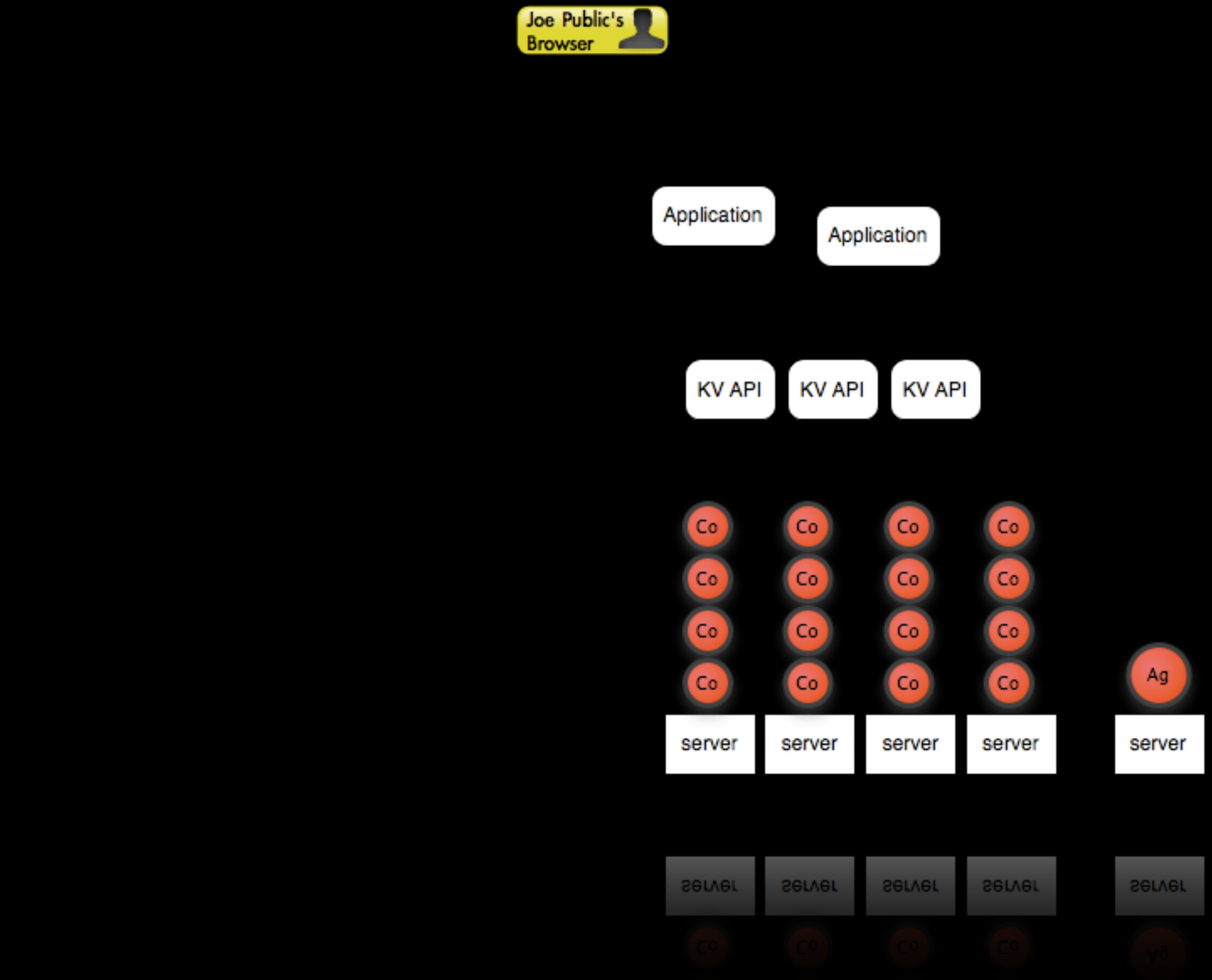
Operational sharding



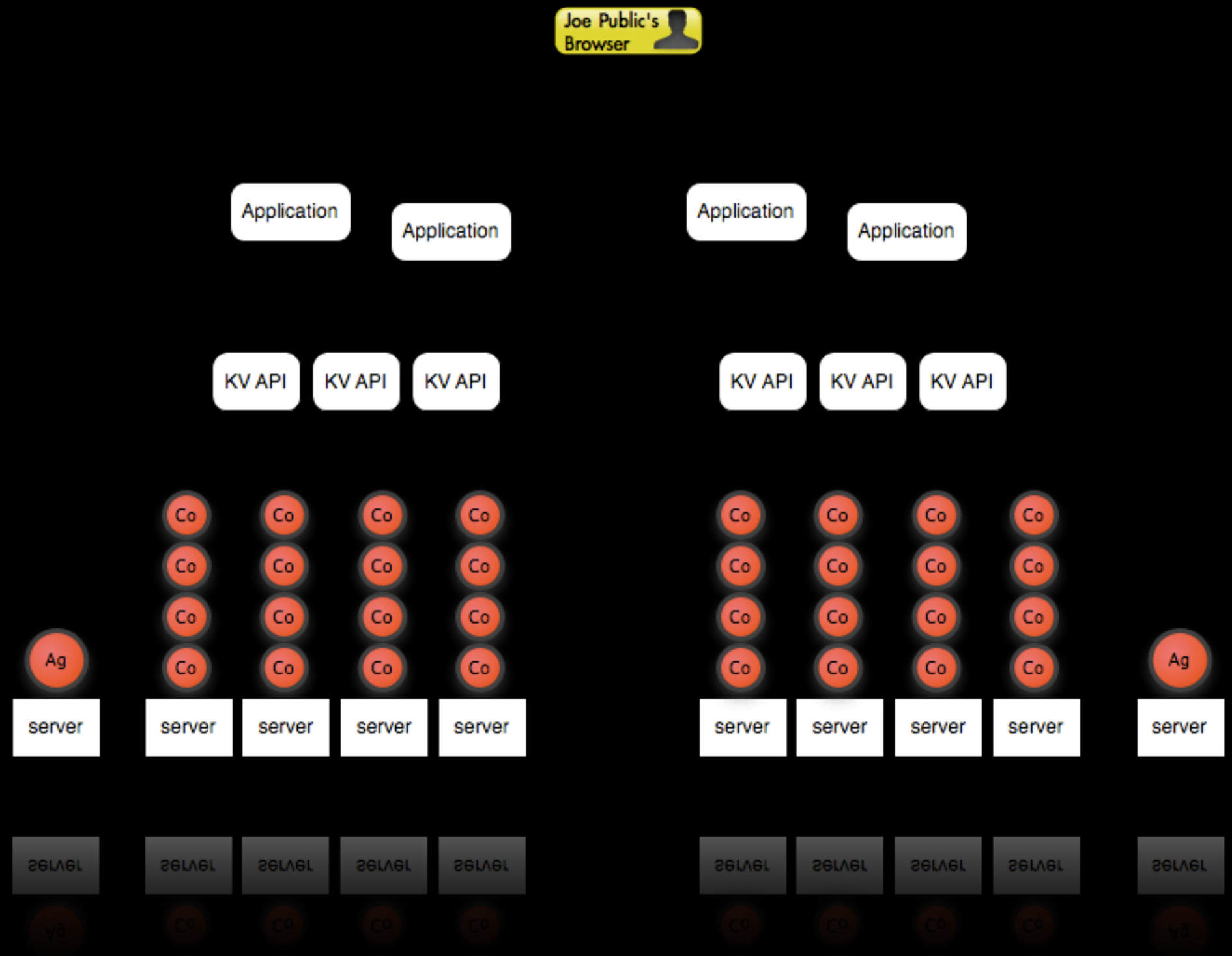
Developers' contract



Ops expandability


Joe Public's Browser
Application
Application
KV API
KV API
KV API
Co
Ag
server

Views



Active-active DCs

Uses of the KV store

- Homepage - 40M users' preferences
- /spaces* - preferences
- iPlayer - 2-3M playlists
- LabUK* experiments

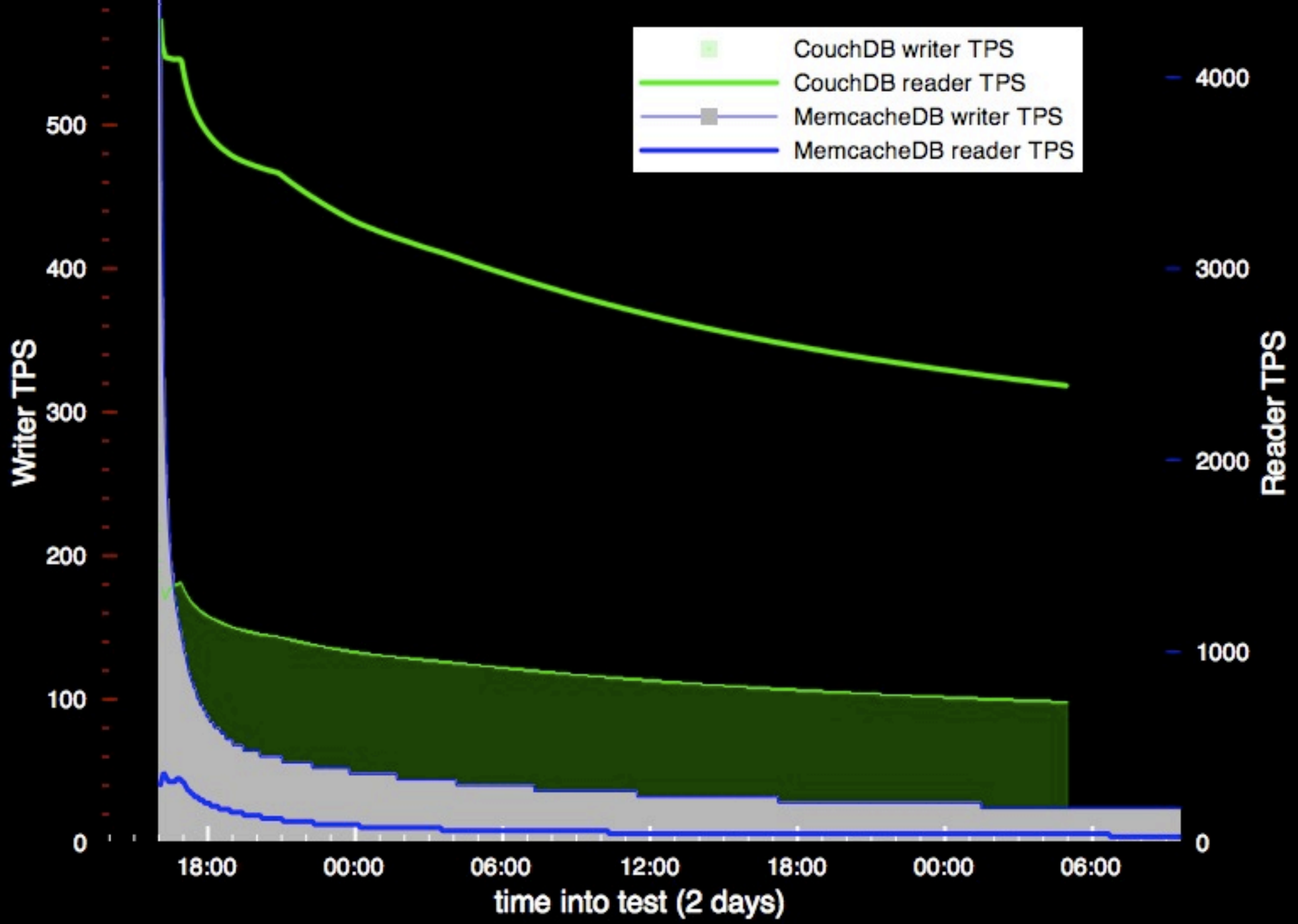
- These are all in the works, coming in the next 3-6 months ... or have had their identities changed

Why the KV API?

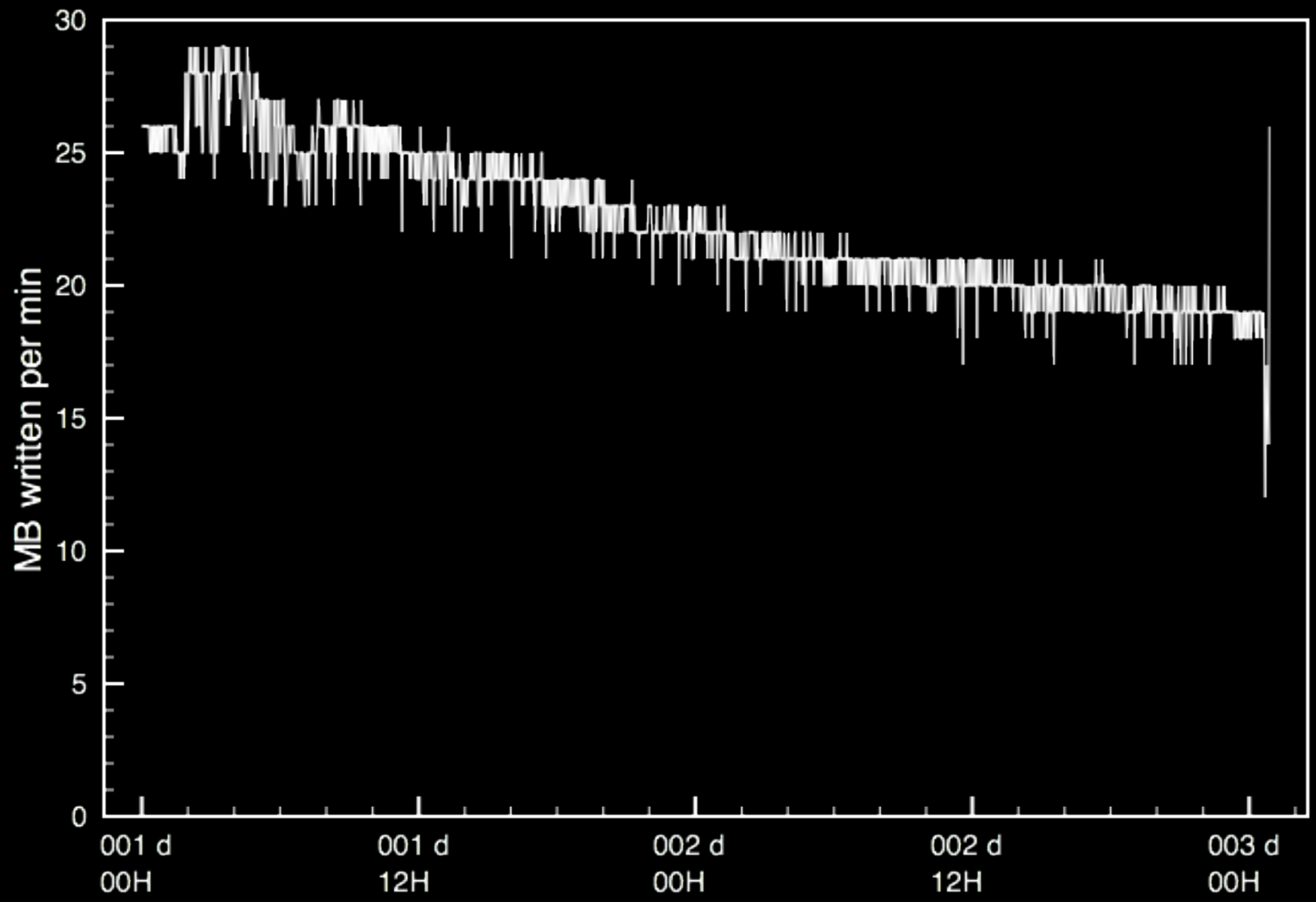
- API stability
- Contract with the developer community
- Developer usage constraints
- Business-level access control
- Monitoring and alerting
- Expandability

Why CouchDB?

- Append-only file
- Good IO to disk
- Graceful performance degredation under load
- Consistent memory usage
- CouchDB + Erlang: OS managability


CouchDB writer TPS
CouchDB reader TPS
MemcacheDB writer TPS
MemcacheDB reader TPS
Writer TPS
500
400
300
200
100
0
Reader TPS
4000
3000
2000
1000
0
18:00
00:00
06:00
12:00
18:00
00:00
06:00
time into test (2 days)

Benchmarks


MB written per min
30
25
20
15
10
5
0
001 d
00H
001 d
12H
002 d
00H
002 d
12H
003 d
00H

Benchmarks

Hardware: our CouchDB servers

- 2 x Quad core Intel Xeon
- 2 GHz
- 16 GB RAM
- 4 x 10k SAS RAID 5
- 0.5 TB usable HDD

Disk IO bound

- The hardware is our limitation
- beam.smp works well in our setup
- CPU interrupts, load spread quite evenly

What next?

- Relying on hardware redundancy - not software :-(
- Talk to the Meebo folks (worried about Twisted)

- Better understanding of compaction, near continuous replication and conflicts
- https://monitor.forge.bbc.co.uk/zport/dmd/Reports/Multi-Graph%20Reports/KV%20store

Thank you

BBC Feeds Hub

26th June 09

BBC Audio & Music Interactive and Mobile

LShift

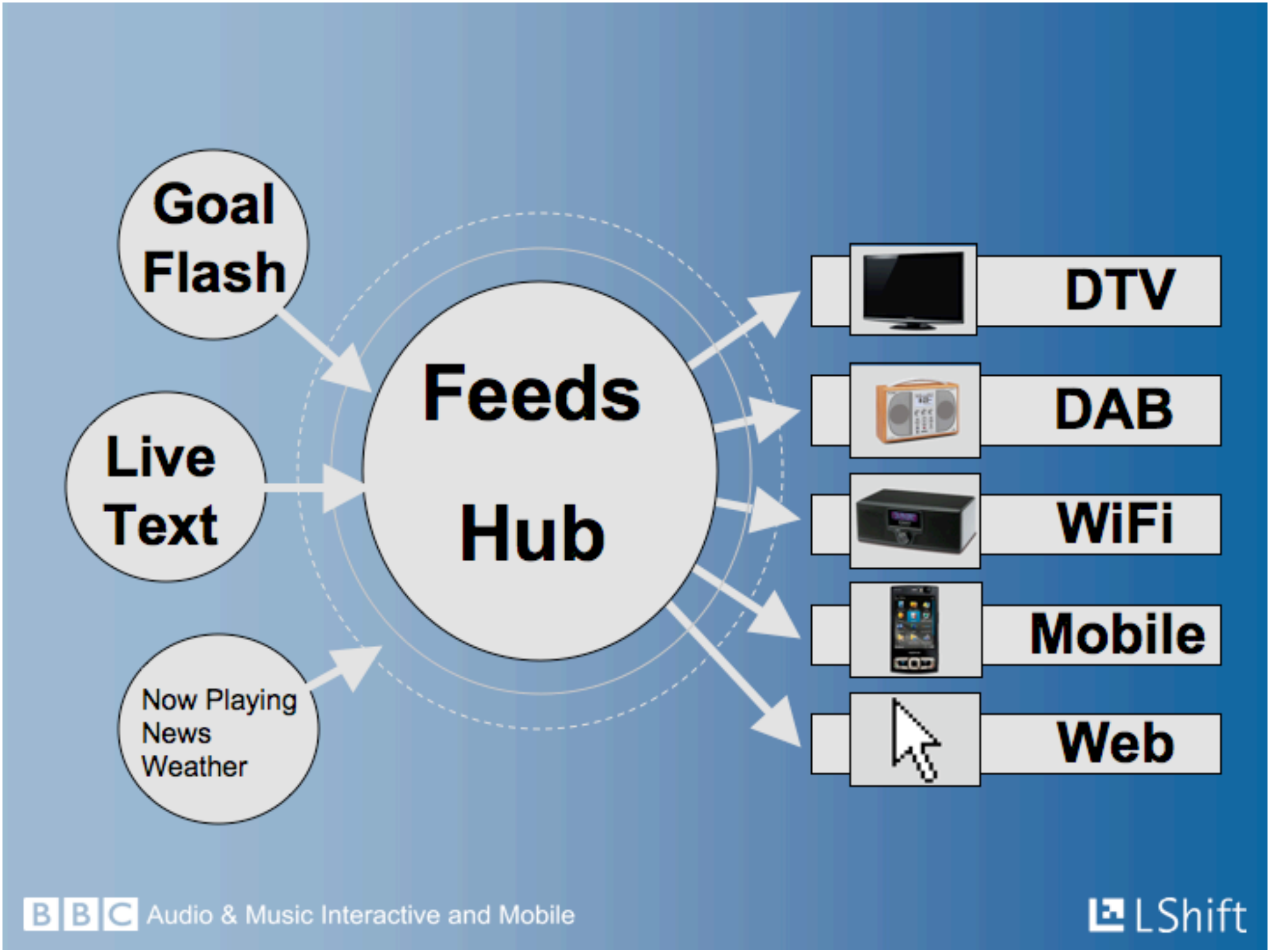
CHALLENGE

BBC Audio & Music Interactive and Mobile

LShift

"The number of new projects across the BBC starting to use feeds in creative ways is growing very quickly - just think of spaghetti... on a massive scale."


1328.


Goal Flash
Live Text
Now Playing
News
Weather
Feeds Hub
DTV
DAB
WiFi
Mobile
Web
BBC Audio & Music Interactive and Mobile
LShift

DAB Live Text:
Man Utd 1 – 0 Tottenham

Mobile:
Man U 1 – Spurs 0

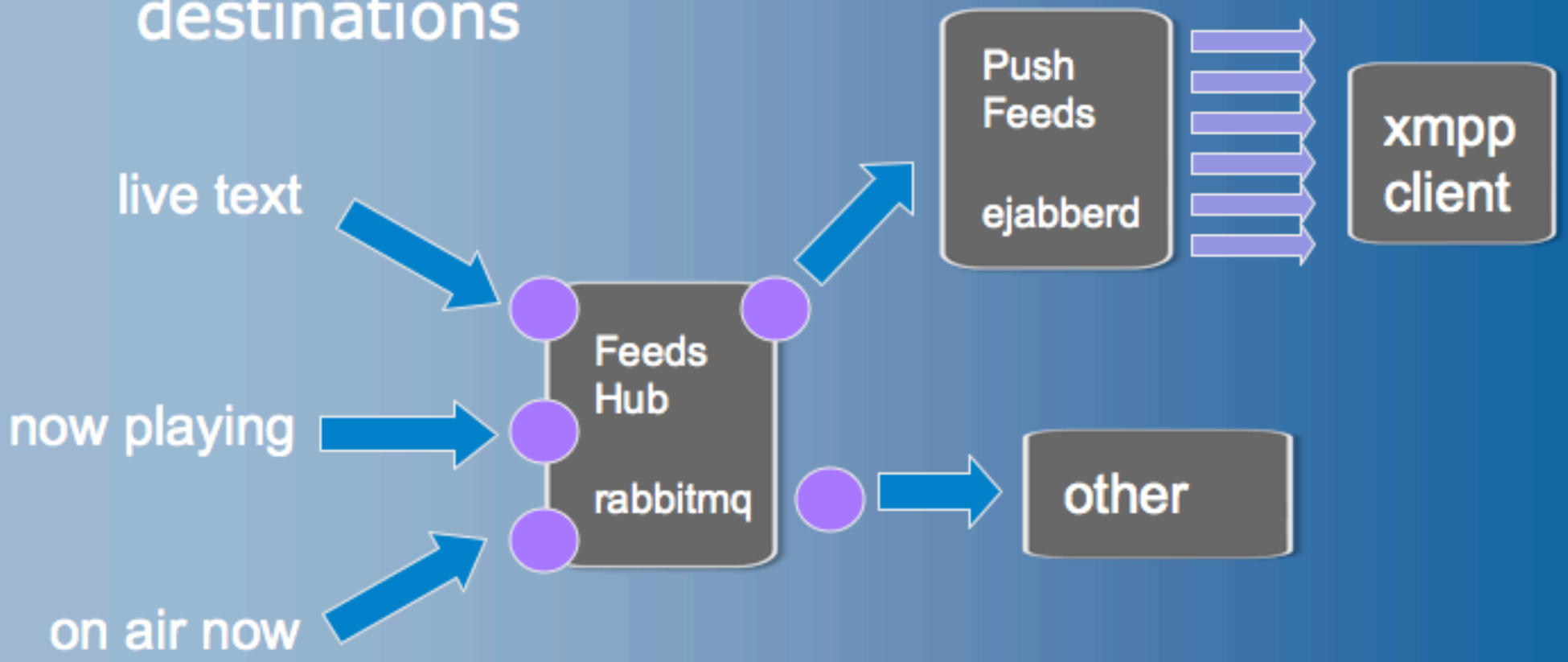
Web:
Manchester United 1 – 0 Tottenham Hotspur

MODEL

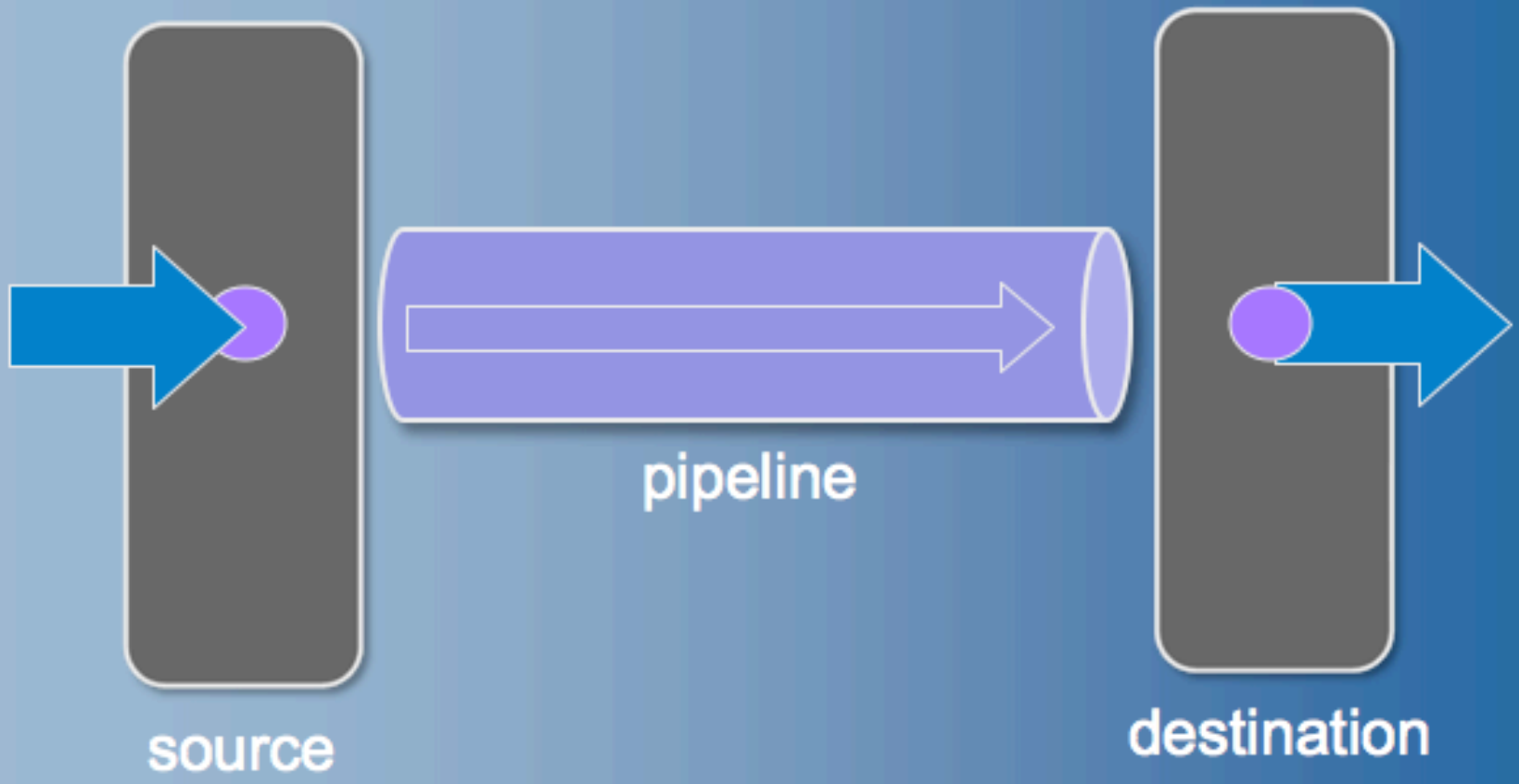
BBC Audio & Music Interactive and Mobile

LShift

Messages from various sources are collated, transformed and routed to their destinations

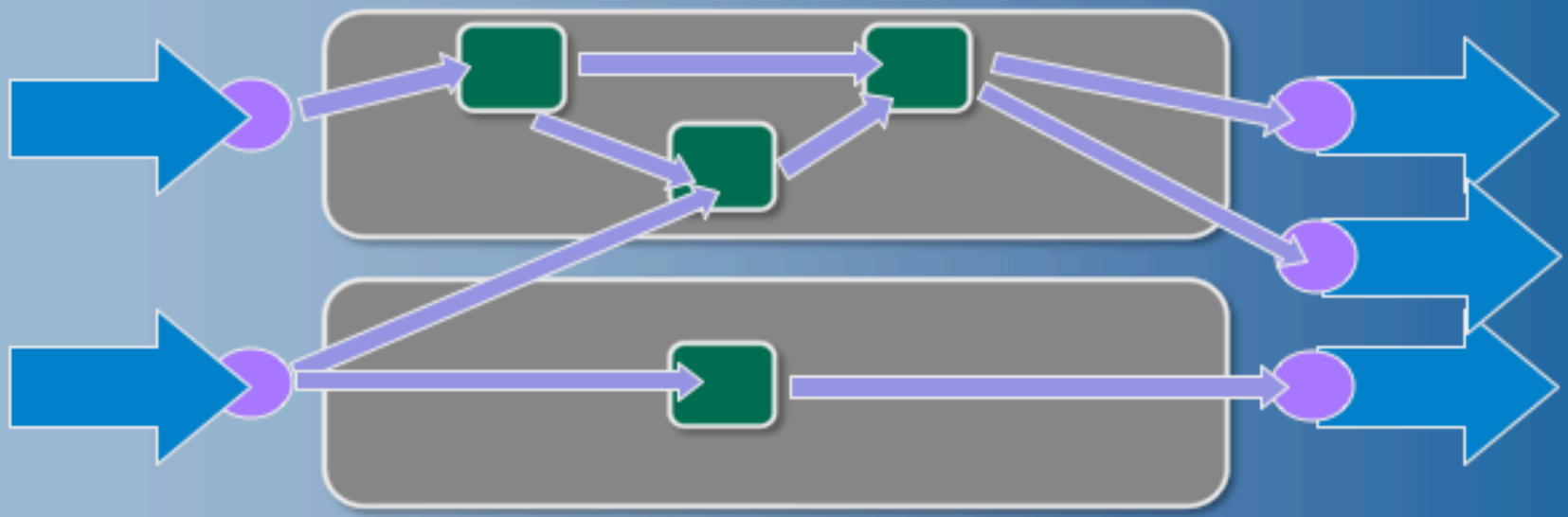


Messages arrive at SOURCES
and arrive at DESTINATIONS
via a PIPELINE

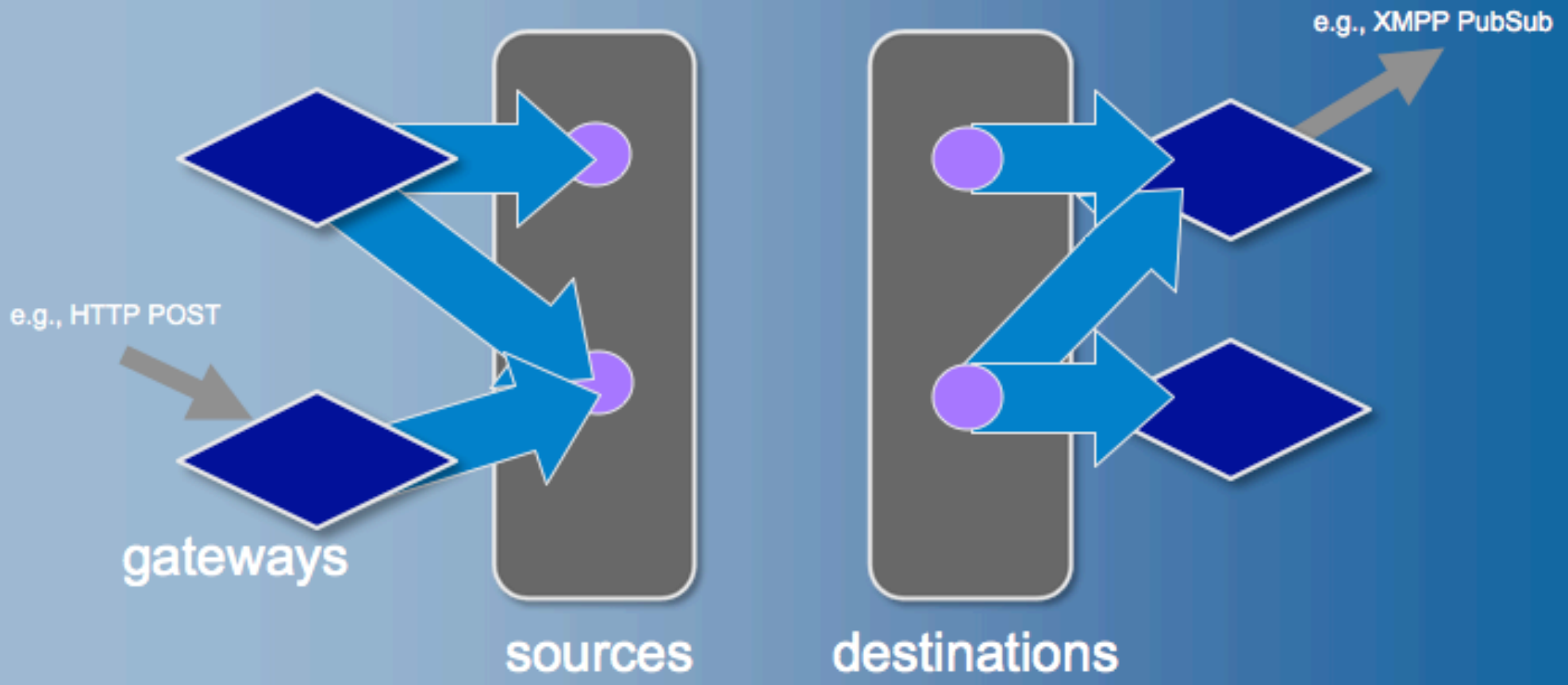


PIPELINES are connected COMPONENTS expressing business rules ...

which can all be wired together arbitrarily



GATEWAYS are the relays to outside by which messages arrive at SOURCES and depart from DESTINATIONS

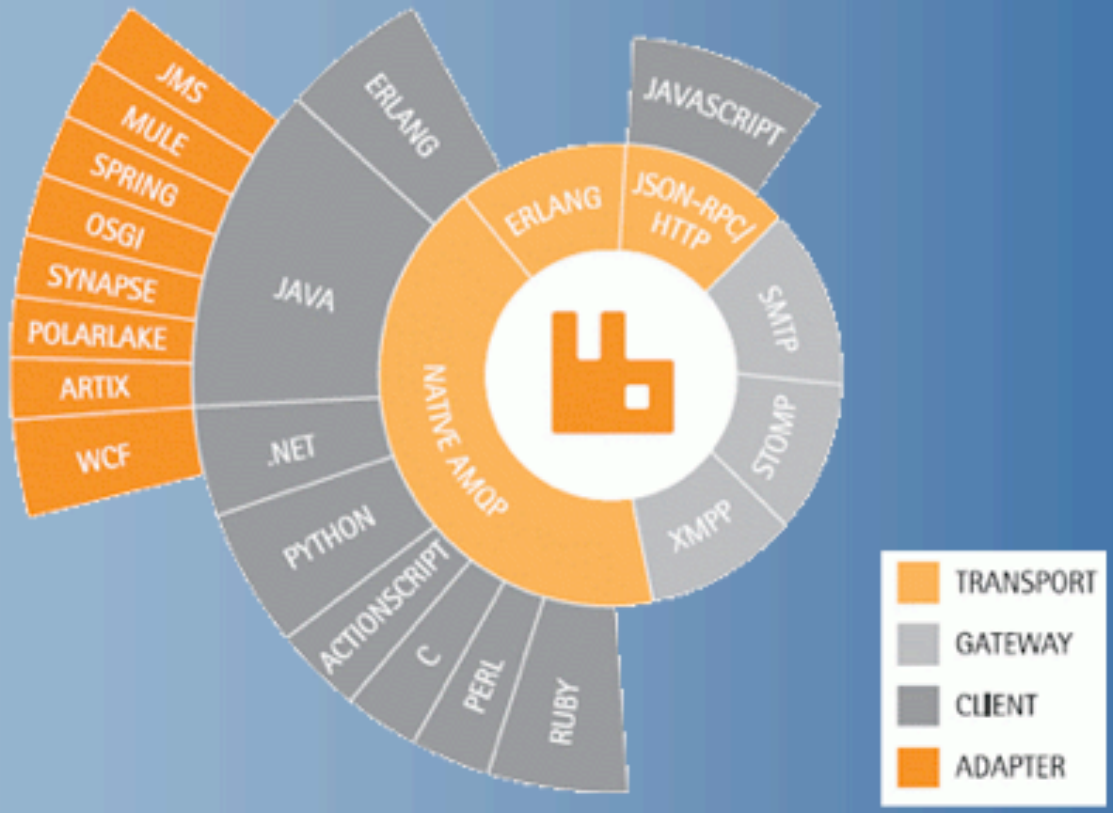


FEEDS HUB ARCHITECTURE

BBC Audio & Music Interactive and Mobile

LShift

RabbitMQ underneath; let others do what they're good at, e.g., ejabberd, RabbitHub



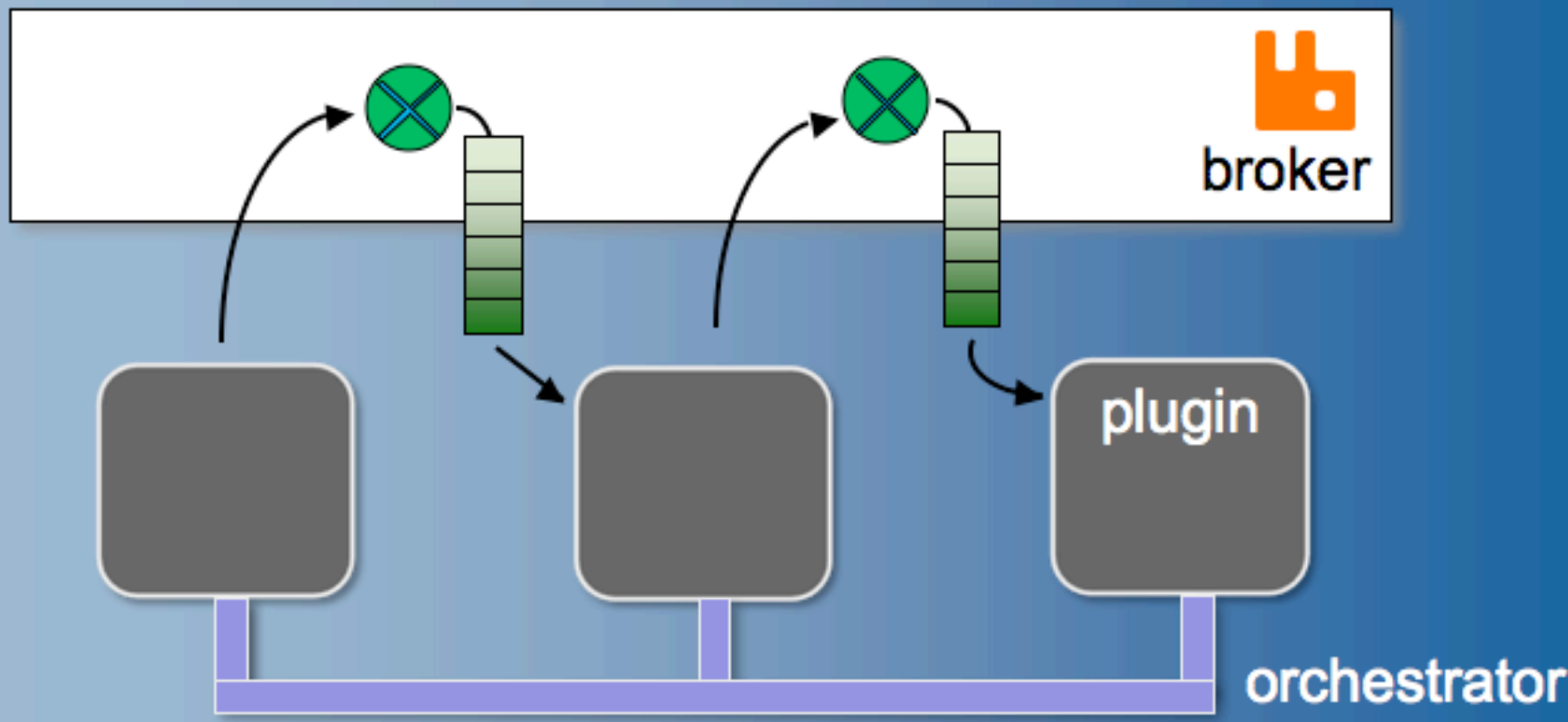
Model items are stored as documents in CouchDB

Model items refer to PLUGINS – eg "XMPP Gateway", "Regular expression replacer"

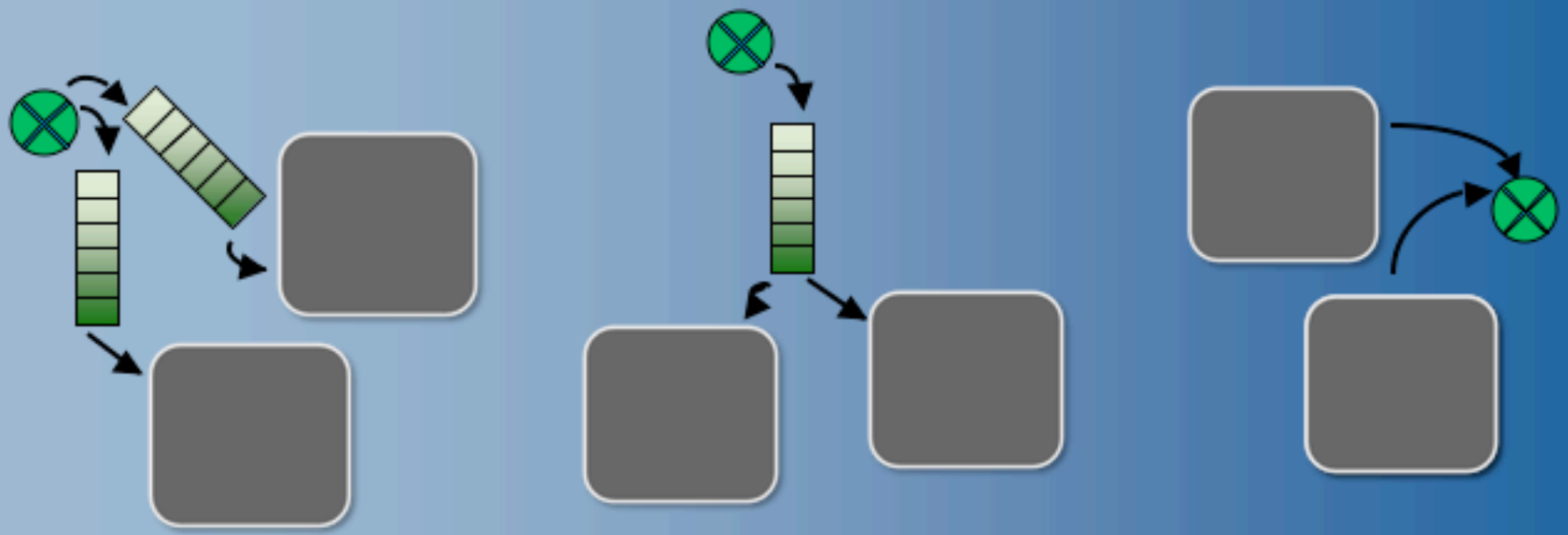
Plugins are just programs

The ORCHESTRATOR manages a process for each model item

It wires everything together with AMQP exchanges and queues



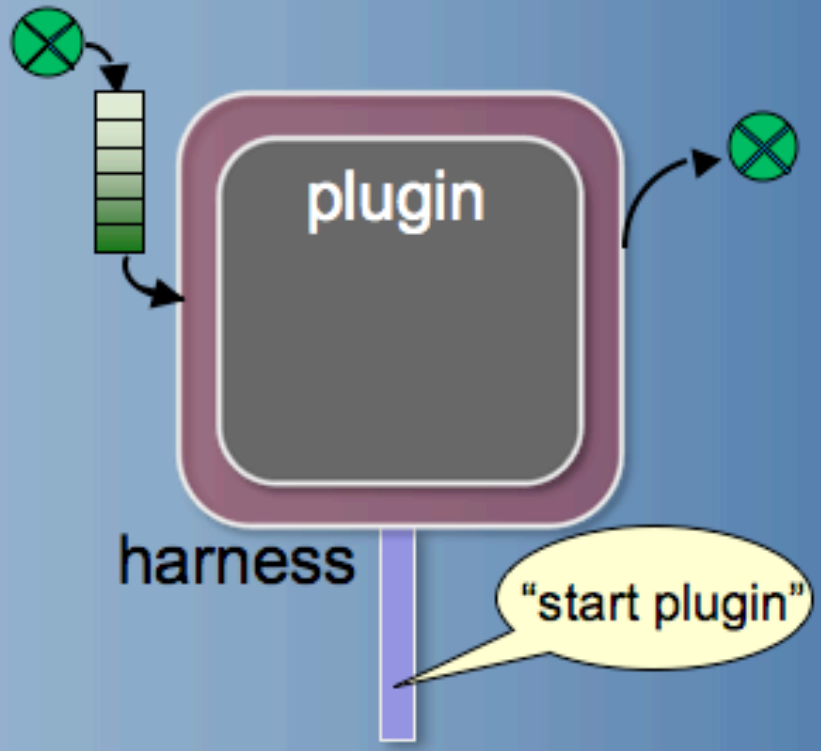
Streams can be interleaved, duplicated, and load-balanced using the messaging layer



Duplicate

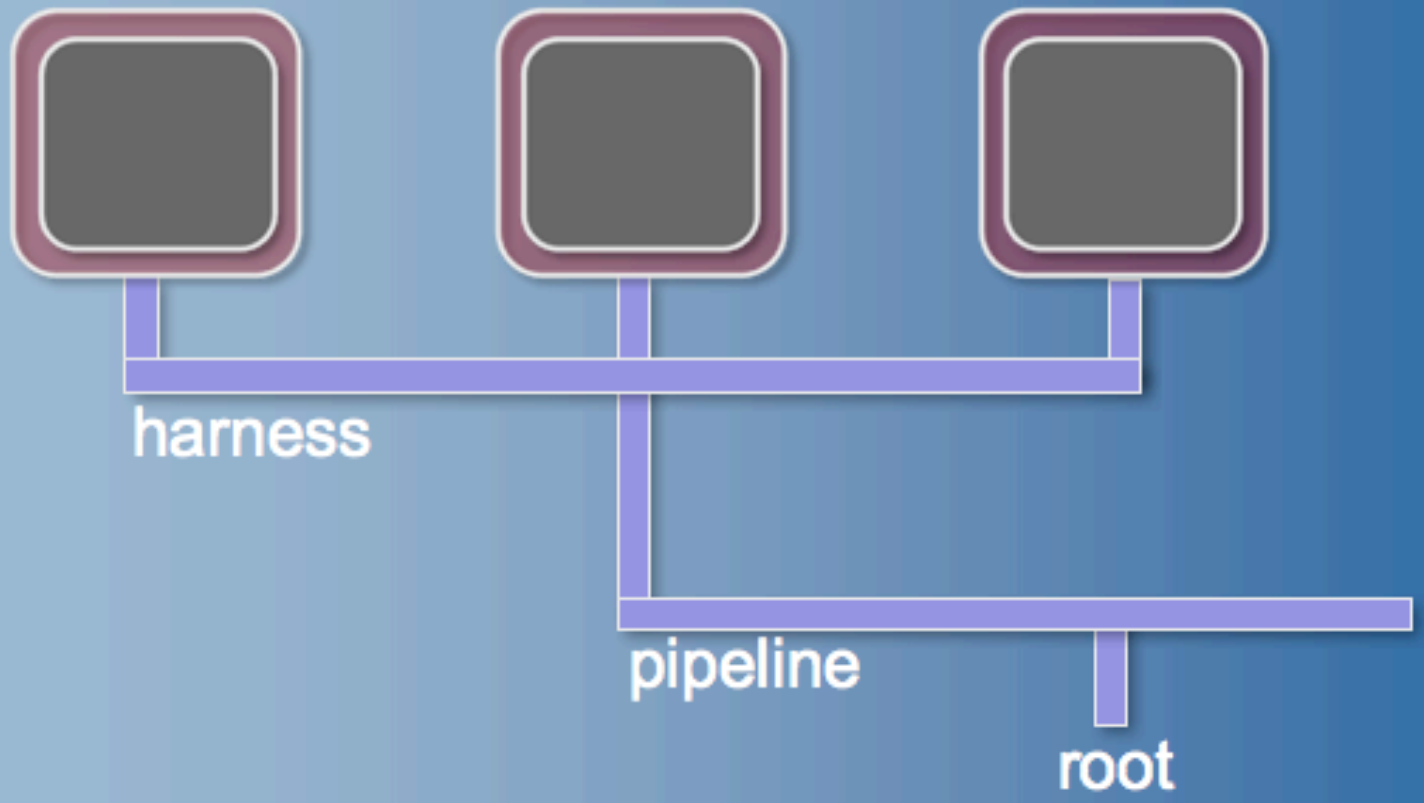
Load-balance

Interleave



Harnesses mediate between the orchestrator and the plugins' programming language environments

Processes are organised into a hierarchy and can be restarted individually by the orchestrator



Open Source

- Feeds Hub
  - Prototype release scheduled 6th July
  - http://www.bbc.co.uk/blogs/radiolabs/2009/04/introducing_bbc_feeds_hub.shtml
- RabbitMQ
  - www.rabbitmq.com
  - http://groups.google.com/group/rabbitmq-discuss