Cloud is such stuff as dreams are made on

Patrick Chanezon
Developer Advocate, Cloud & Apps
chanezon@google.com
http://twitter.com/chanezon

Paris JUG
July 2011

P@ in a nutshell

accenture

AOL

Sun microsystems

Google™

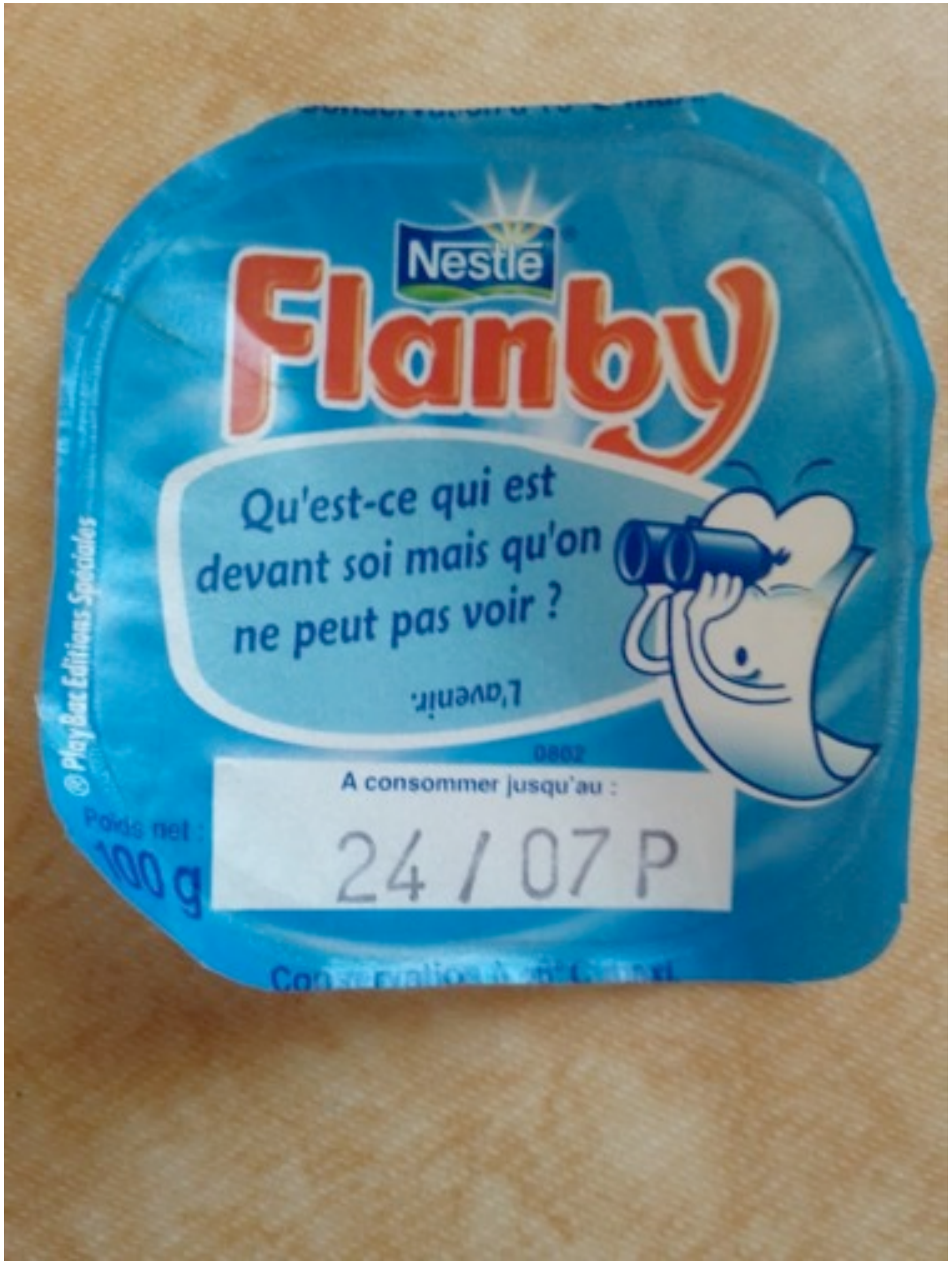
- French, based in San Francisco
- Developer Advocate, Google Cloud & Apps
- Software Plumber, API guy, mix of Enterprise and Consumer
    - 18 years writing software, backend guy with a taste for javascript
    - 2 y Accenture (Notes guru), 3 y Netscape/AOL (Servers, Portals), 5 y Sun (ecommerce, blogs, Portals, feeds, open source)
- 6 years at Google, API guy (first hired, helped start the team)
    - Adwords, Checkout, Social, HTML5, Cloud

Predictions

“The future is already here — it's just not very evenly distributed”

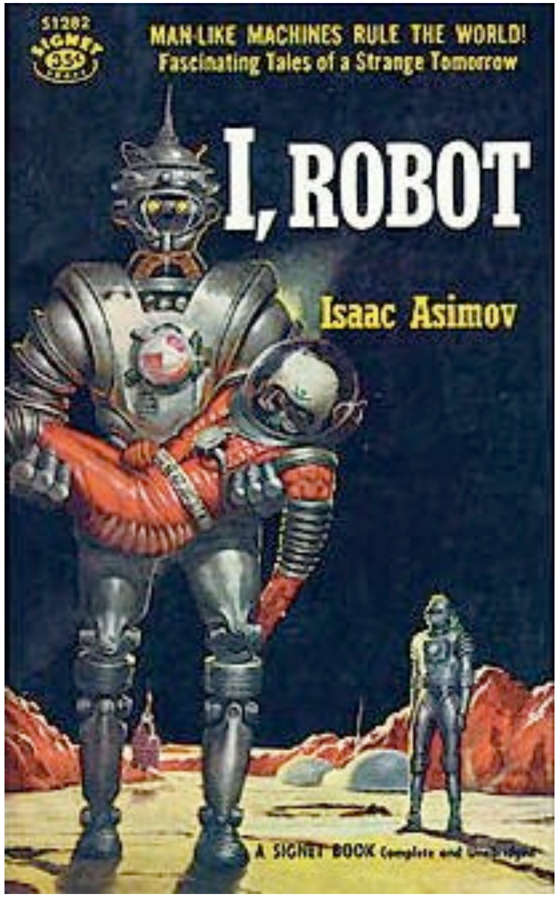
William Gibson

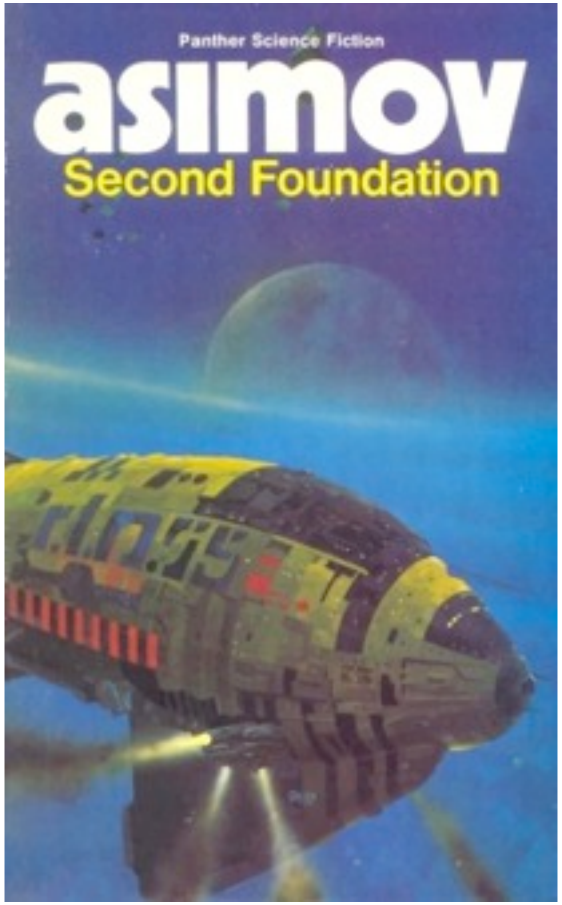
Predictions


Nestlé
Flanby
Qu'est-ce qui est devant soi mais qu'on ne peut pas voir ?
L'avenir.
® PlayBac Editions Spéciales
0802
A consommer jusqu'au :
Poids net :
100 g
24 / 07 P

En VF:-)

Dreams Of my childhood

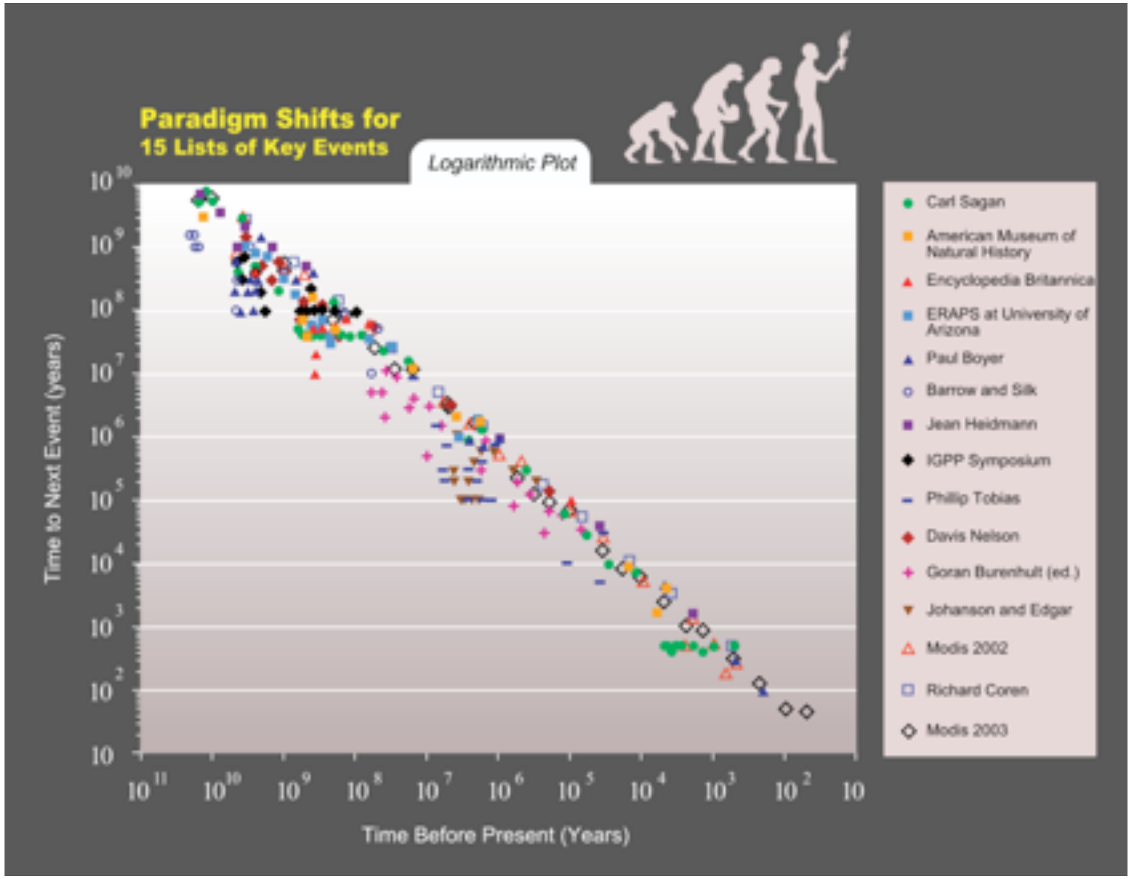

51282
SIGNET
35¢
MAN-LIKE MACHINES RULE THE WORLD!
Fascinating Tales of a Strange Tomorrow
I, ROBOT
Isaac Asimov
A SIGNET BOOK Complete and Unabridged



Accelerando / Singularity, in a Galaxy far far away

- Even if we automate ourselves out of a job every 10 years
- ...I don’t think the singularity is near!

HUGO AWARD-WINNING
AUTHOR OF GLASSHOUSE
charles stross
"Accelerando is to cyberpunk what Napster was to the music industry: volatile, visionary, a bit flawed, and a lot of fun."
—Entertainment Weekly
accelerando

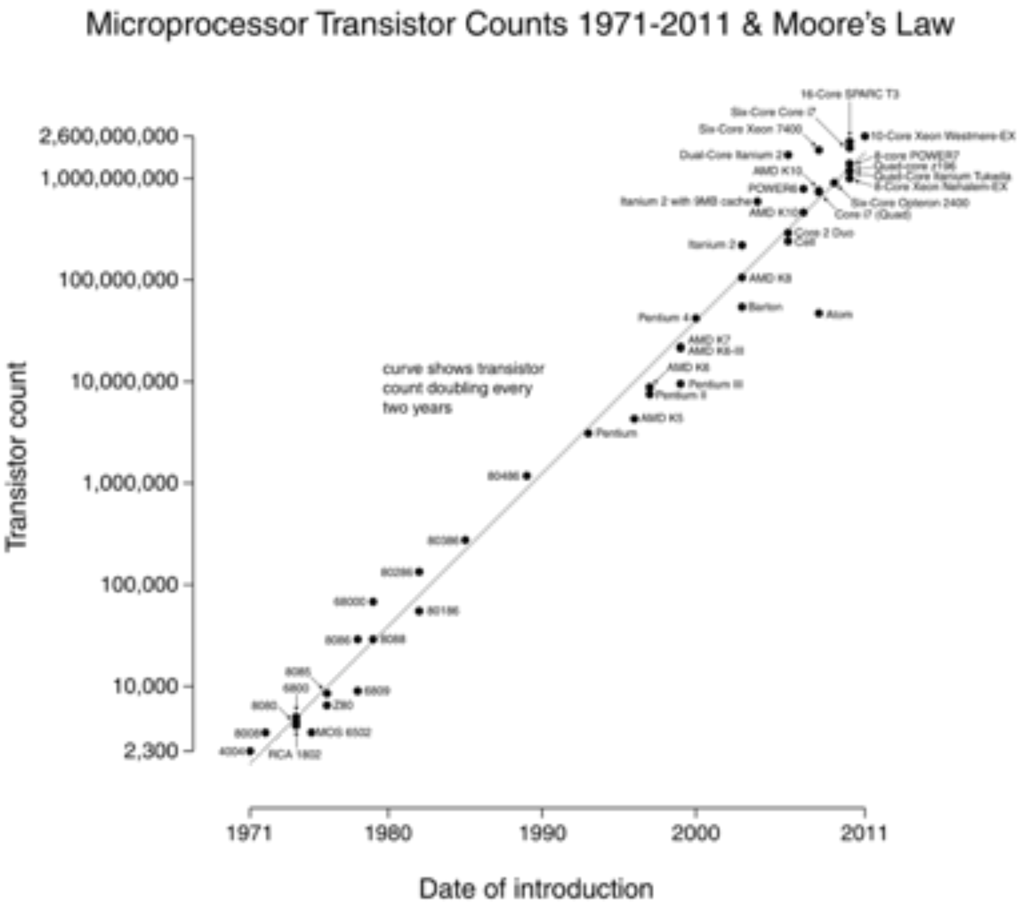


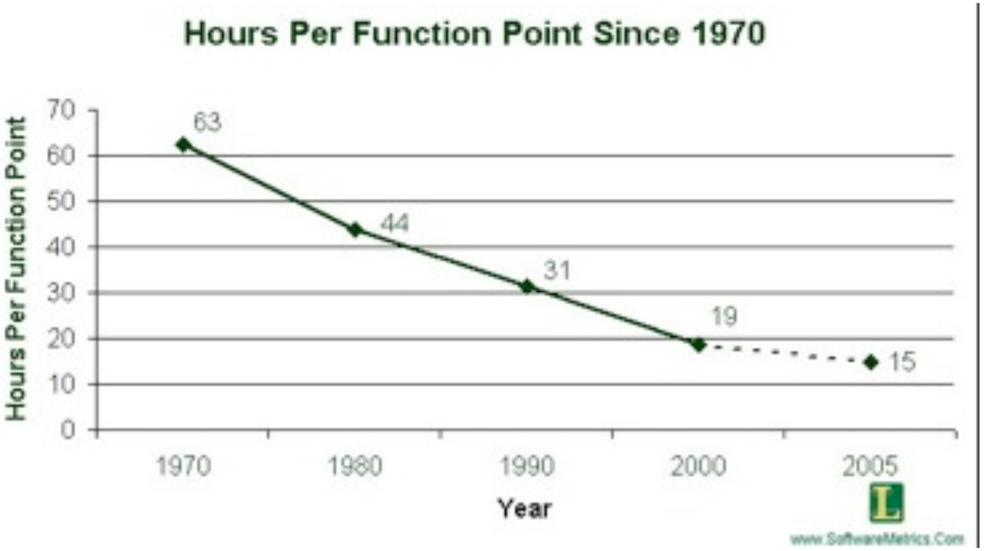
Moore's Law is for Hardware Only

- Does not apply to software

- Productivity gains not keeping up with hardware and bandwidth

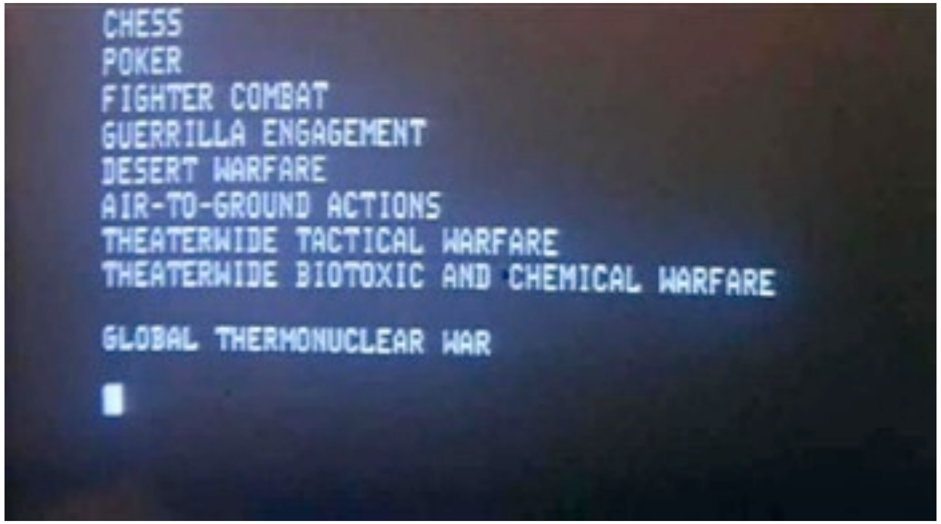
- Writing software is hard, painful, and still very much a craft





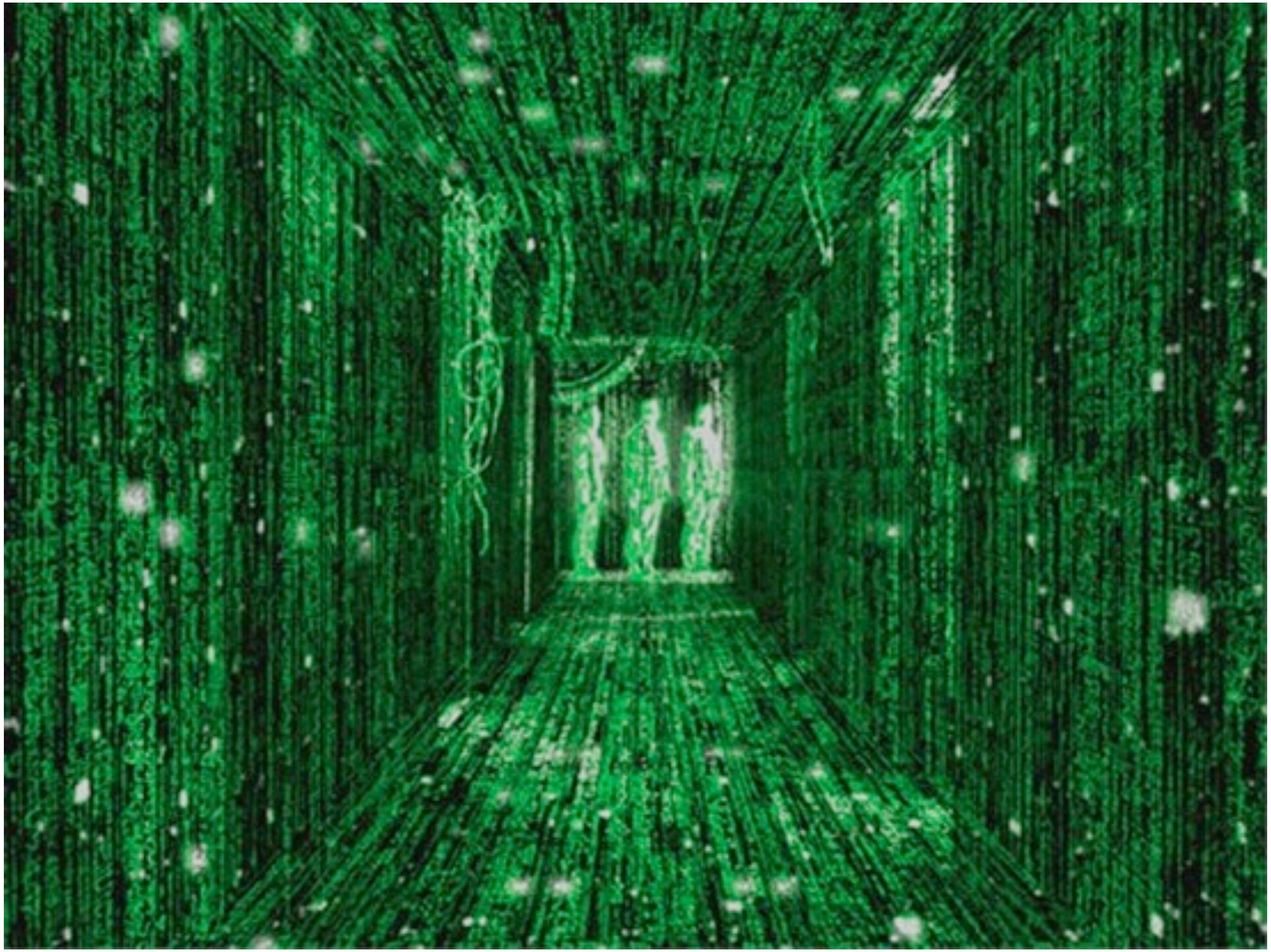
Architecture Changes: 60’s Mainframe

Architecture Changes: 80’s Client-Server

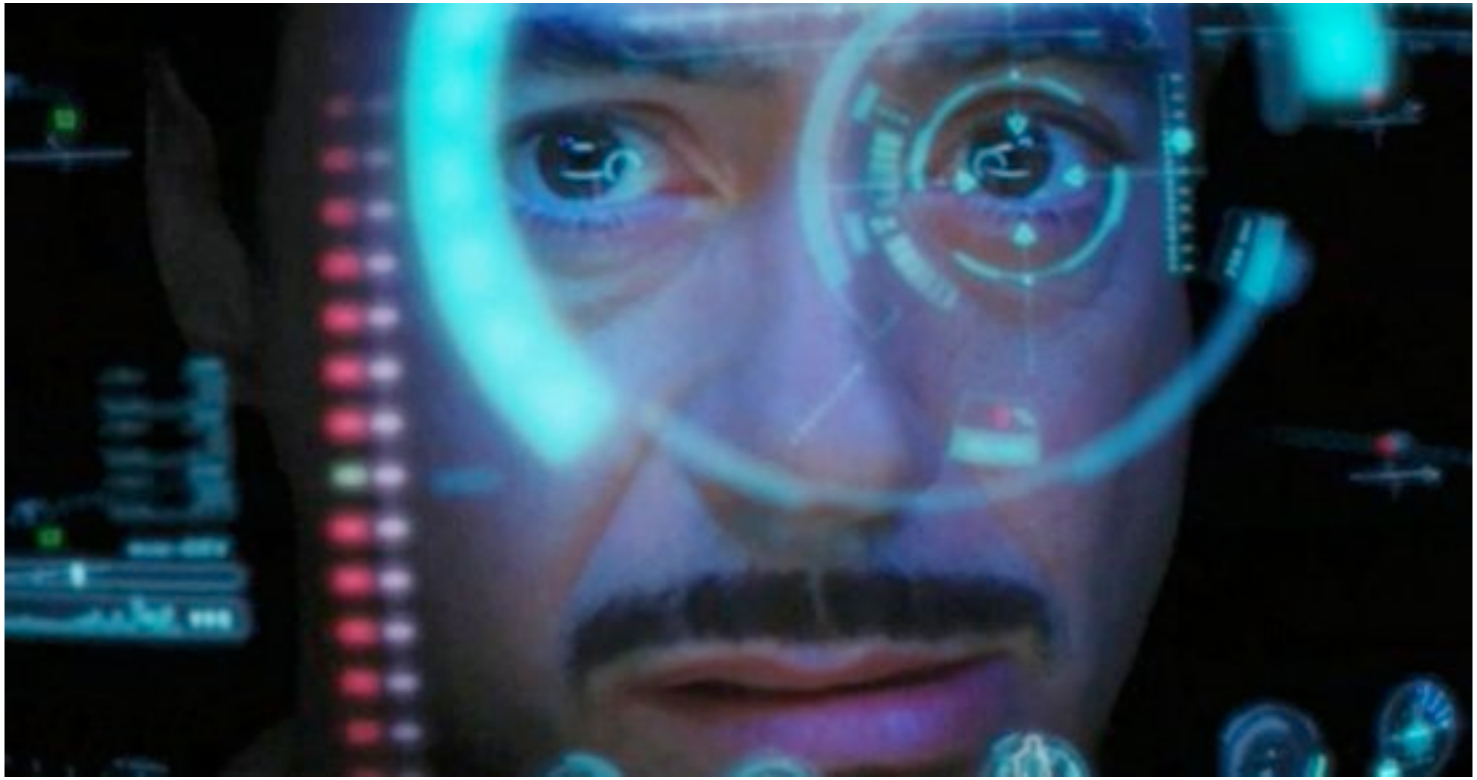


```
CHESS
POKER
FIGHTER COMBAT
GUERRILLA ENGAGEMENT
DESERT WARFARE
AIR-TO-GROUND ACTIONS
THEATERWIDE TACTICAL WARFARE
THEATERWIDE BIOTOXIC AND CHEMICAL WARFARE

GLOBAL THERMONUCLEAR WAR
```

Architecture Changes: 90’s Web



Architecture Changes: 2010's Cloud, HTML5, Mobile

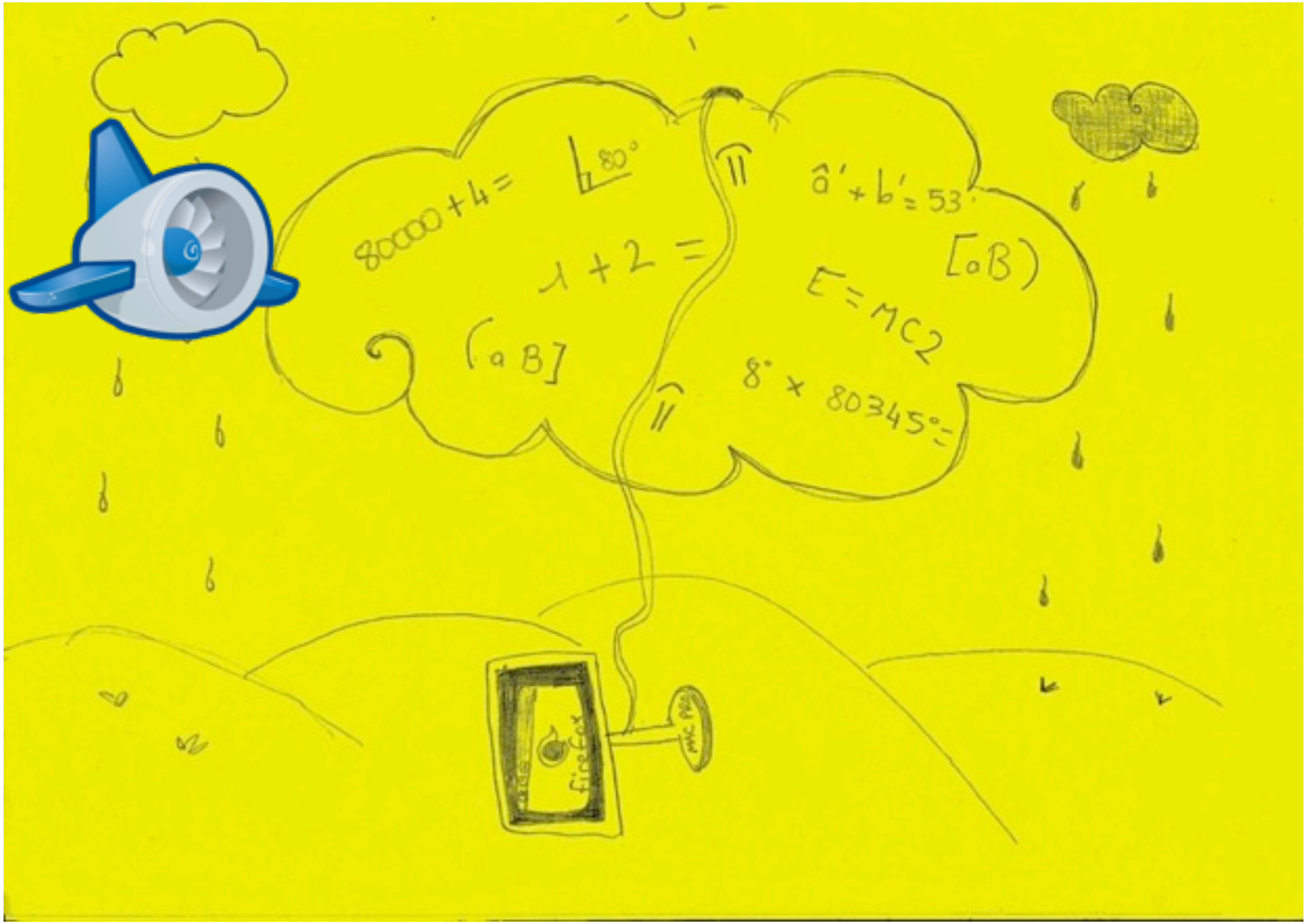


Back to Client Server: Groovy Baby!

Other components of change

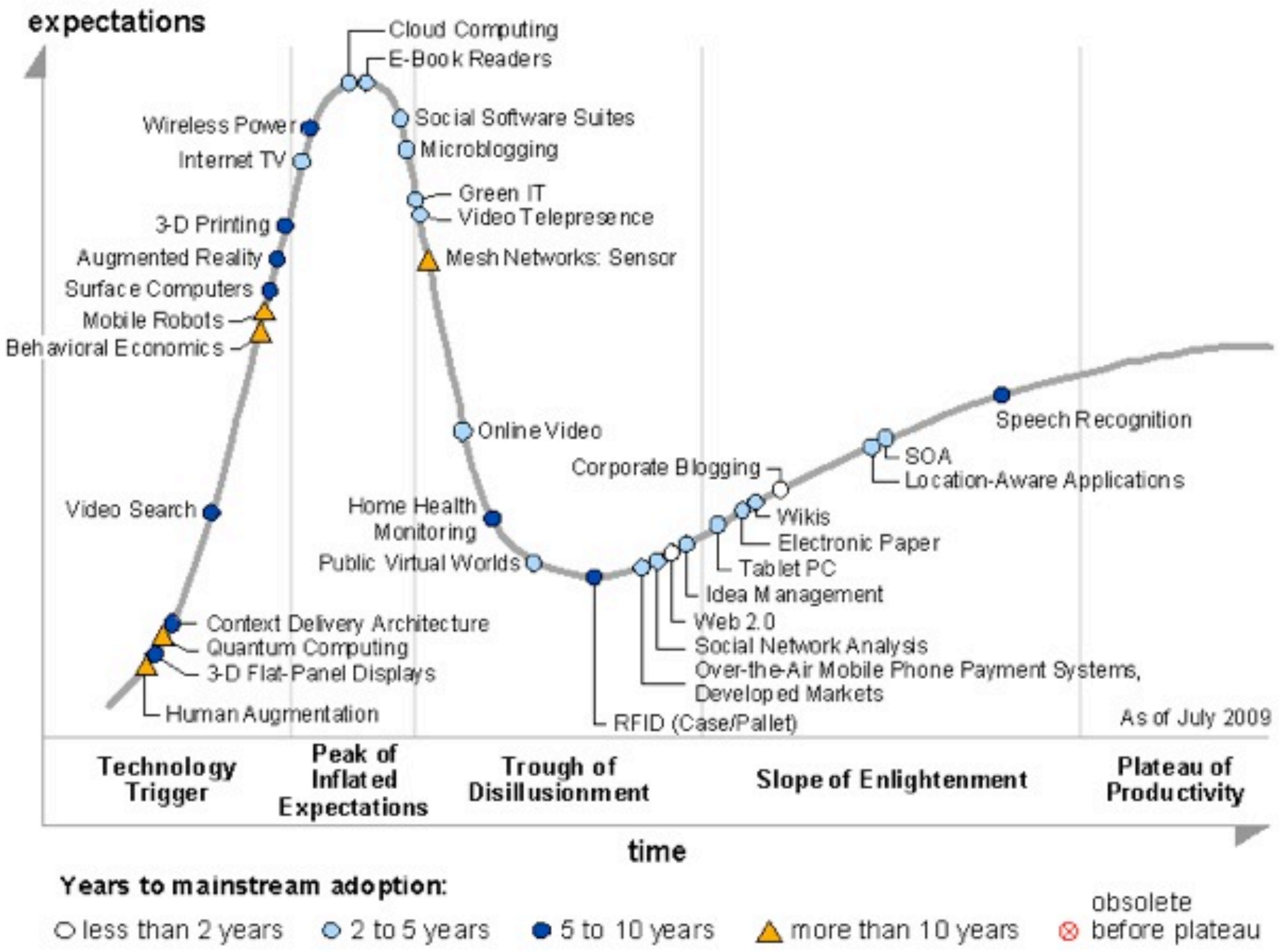
- Client: Browsers, Mobile
- Server: Web services, apis, rest and ajax
- Services: Social, Geo

This talk is about the Server Side, the Cloud



Cloud, according to my daughter Eliette

Hype warning: Cloudy, with a chance of real innovation



Source: Gartner (August 2009)

Cloud started at Consumer websites solving their needs

- Google, Amazon, Yahoo, Facebook, Twitter
- Large Data Sets
- Storage Capacity growing faster than Moore’s Law
- Fast Networks
- Horizontal -> Vertical scalability
- Open Source Software
- Virtualization
- Cloud is a productization of these infrastructures
    - Public Clouds Services: Google, Amazon
    - Open Source Software: Hadoop, Eucalyptus, Cloud Foundry

Factors Driving Cloud Adoption

- Technical
- Economic
- Cultural

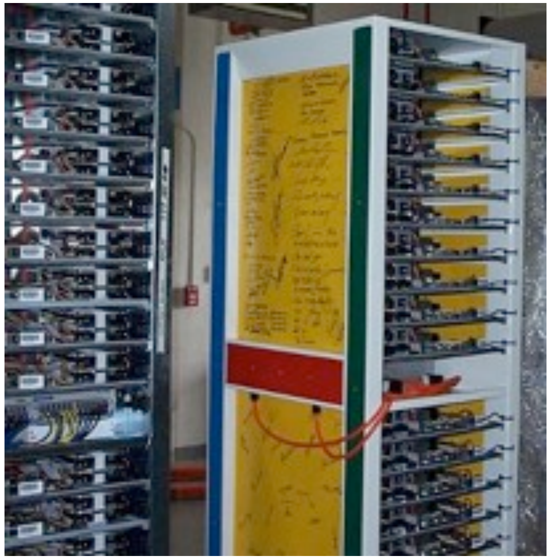
Infrastructure culture

- Larry and Serguey’s 1998 paper ”The Anatomy of a Large-Scale Hypertextual Web Search Engine"
  - http://infolab.stanford.edu/~backrub/google.html
- Other Google Research papers since then
  - http://research.google.com/pubs/papers.html
- Build on the shoulders of giants
- Custom stack made of standards parts: machines, linux, servers
- Standard infrastructure: sharding, GFS, MapReduce, BigTable
- Google App Engine: easy cloud, for Googlers and others developers
- Standard languages: c/c++, java, python
- Horizontal scalability: parallel and asynchronous whenever possible

Programming the Cloud – The Google Way

- Fault tolerant distributed storage: Google File System
- Distributed shared memory: Bigtable
- New programming abstractions: MapReduce
- Domain Specific Languages: Sawzall

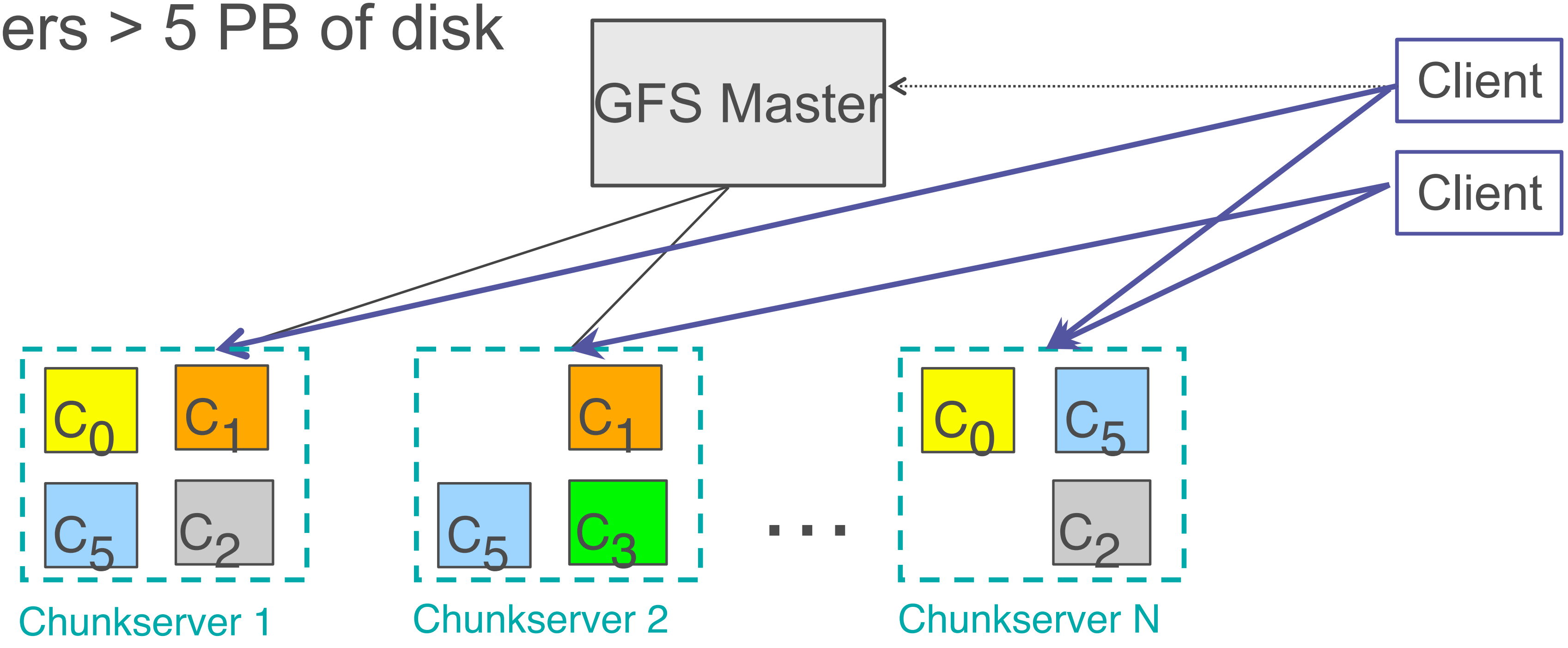
Google.stanford.edu (Circa 1997)



Current Rack Design

Fault Tolerant Distributed Disk Storage: GFS

- Data replicated 3 times. Upon failure, software re-replicates.
- Master: Manages file metadata. Chunk size 64 MB.
- Optimized for high-bandwidth sequential read / writes
- Clusters > 5 PB of disk



http://research.google.com/archive/gfs-sosp2003.pdf

Distributed Shared Memory: Bigtable

- Sparse, distributed, persistent, multidimensional, sorted
- Not a relational database (RDBMS): no schema, no joins, no foreign key constraints, no multi-row transactions
- Each row can have any number of columns, similar to a dictionary data structure for each row.
- Basic data types: string, counter, byte array
- Accessed by row key, column name, timestamp
- Data split into tablets for replication
- Largest cells are > 700TB

http://research.google.com/archive/bigtable-osdi06.pdf

Datastore layers

	Complex queries	Entity Group Transactions	Queries on properties	Key range scan	Get and set by key
Datastore	✓	✓	✓	✓	✓
Megastore		✓	✓	✓	✓
Bigtable				✓	✓

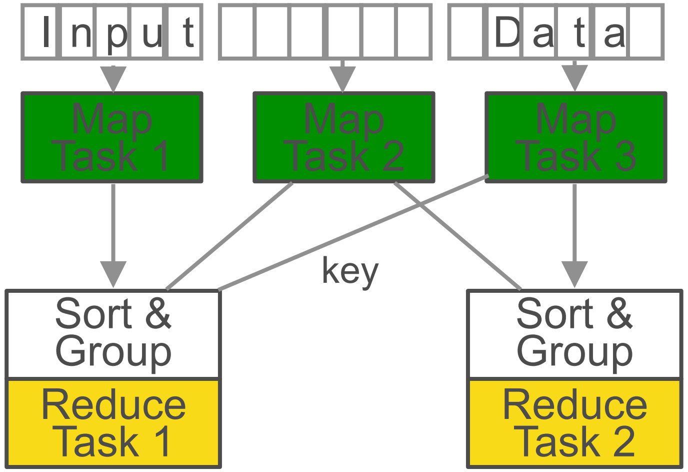
Megastore API

- “Give me all rows where the column ‘name’ equals ‘ikai’”
- “Transactionally write an update to this group of entities”
- “Do a cross datacenter write of this data such that reads will be strongly consistent” (High Replication Datastore)
- Megastore paper: http://www.cidrdb.org/cidr2011/Papers/CIDR11_Paper32.pdf

Programming Abstraction: MapReduce

- Represent problems as Map and Reduce step (inspired by functional programming)
- Distribute data among many machines, execute same computation at each machine on its dataset
- Infrastructure manages parallel execution
- Open source implementation: Hadoop

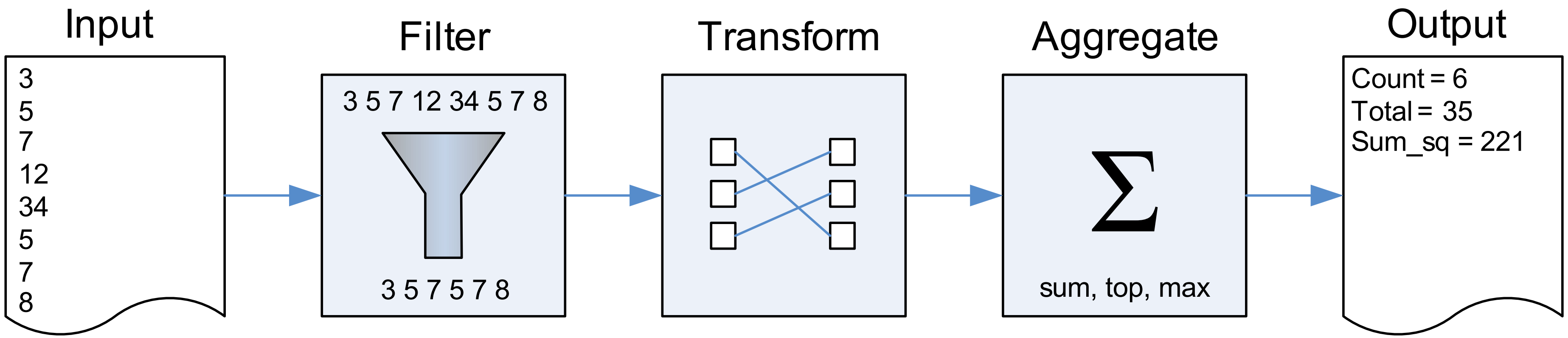
```
map(in_key, data)
 → list(key, value)

reduce(key, list(values))
→ list(out_data)
```



http://research.google.com/archive/mapreduce.html

Language for Parallel Log Processing: Sawzall

- Commutative and associative operations allow parallel execution and aggregation
- Language avoids specifying order by replacing loops with quantifiers (constraints)



Input

3
5
7
12
34
5
7
8

Filter

3 5 7 12 34 5 7 8

3 5 7 5 7 8

Transform

Aggregate

Σ

sum, top, max

Output

Count = 6
Total = 35
Sum_sq = 221

```
count: table sum of int;
total: table sum of float;
x: float = input;
emit count <- 1;
emit total <- x;
```

```
function(word: string): bool {
  when(i: some int;
         word[i] != word[$-1-i])
    return false;
  return true;
};
```

http://labs.google.com/papers/sawzall.html

Internet as a Platform: The Challenges

Architect’s Dream ☺

- Loosely coupled
- Extensible
- Standards-based
- Fault tolerant
- Unlimited computing power
- Ubiquitous

Internet as a Platform: The Challenges

Architect's Dream

- Loosely coupled
- Extensible
- Standards-based
- Fault tolerant
- Unlimited computing power
- Ubiquitous

Developer's Nightmare

- NO Call Stack
- NO Transactions
- NO Promises
- NO Certainty
- NO Ordering Constraints

New Game Rules

ACID (before)

ACID (today)

New Game Rules

ACID (before)

- Atomic

ACID (today)

New Game Rules

ACID (before)

- Atomic
- Consistent

ACID (today)

New Game Rules

ACID (before)

- Atomic
- Consistent
- Isolated

ACID (today)

New Game Rules

ACID (before)

- Atomic
- Consistent
- Isolated
- Durable

ACID (today)

New Game Rules

ACID (before)

- Atomic
- Consistent
- Isolated
- Durable

ACID (today)

- Associative

New Game Rules

ACID (before)

- Atomic
- Consistent
- Isolated
- Durable

ACID (today)

- Associative
- Commutative

New Game Rules

ACID (before)

- Atomic
- Consistent
- Isolated
- Durable

ACID (today)

- Associative
- Commutative
- Idempotent

New Game Rules

ACID (before)

- Atomic
- Consistent
- Isolated
- Durable

ACID (today)

- Associative
- Commutative
- Idempotent
- Distributed

New Game Rules

ACID (before)

- Atomic
- Consistent
- Isolated
- Durable

Predictive
Accurate

ACID (today)

- Associative
- Commutative
- Idempotent
- Distributed

Flexible
Redundant

Starbucks Does not Use 2-Phase Commit Either

- Start making coffee before customer pays
- Reduces latency
- What happens if…

Starbucks Does not Use 2-Phase Commit Either

- Start making coffee before customer pays
- Reduces latency
- What happens if…

Customer rejects drink

Coffee maker breaks

Customer cannot pay

Starbucks Does not Use 2-Phase Commit Either

- Start making coffee before customer pays
- Reduces latency
- What happens if…

Customer rejects drink → Remake drink

Coffee maker breaks

Customer cannot pay

Starbucks Does not Use 2-Phase Commit Either

- Start making coffee before customer pays
- Reduces latency
- What happens if…

Customer rejects drink → Remake drink

Coffee maker breaks → Refund money

Customer cannot pay

Starbucks Does not Use 2-Phase Commit Either

- Start making coffee before customer pays
- Reduces latency
- What happens if…

Customer rejects drink → Remake drink

Coffee maker breaks → Refund money

Customer cannot pay → Discard beverage

Starbucks Does not Use 2-Phase Commit Either

- Start making coffee before customer pays
- Reduces latency
- What happens if…

Customer rejects drink	→	Remake drink Retry
Coffee maker breaks	→	Refund money
Customer cannot pay	→	Discard beverage

Starbucks Does not Use 2-Phase Commit Either

- Start making coffee before customer pays
- Reduces latency
- What happens if…

Customer rejects drink	→	Remake drink Retry
Coffee maker breaks	→	Refund money Compensation
Customer cannot pay	→	Discard beverage

Starbucks Does not Use 2-Phase Commit Either

- Start making coffee before customer pays
- Reduces latency
- What happens if…

Customer rejects drink	→	Remake drink Retry
Coffee maker breaks	→	Refund money Compensation
Customer cannot pay	→	Discard beverage Write-off

Commoditization of distributed computing concepts & tools

- Languages: Erlang concepts -> Go, Scala
- NoSQL Zoo: BigTable, HBase, MongoDB, Reddis, Cassandra
- Map/Reduce: Apache Hadoop
- Paxos, Eventual Consistency, CAP Theorem
- REST, statelessness, idempotency

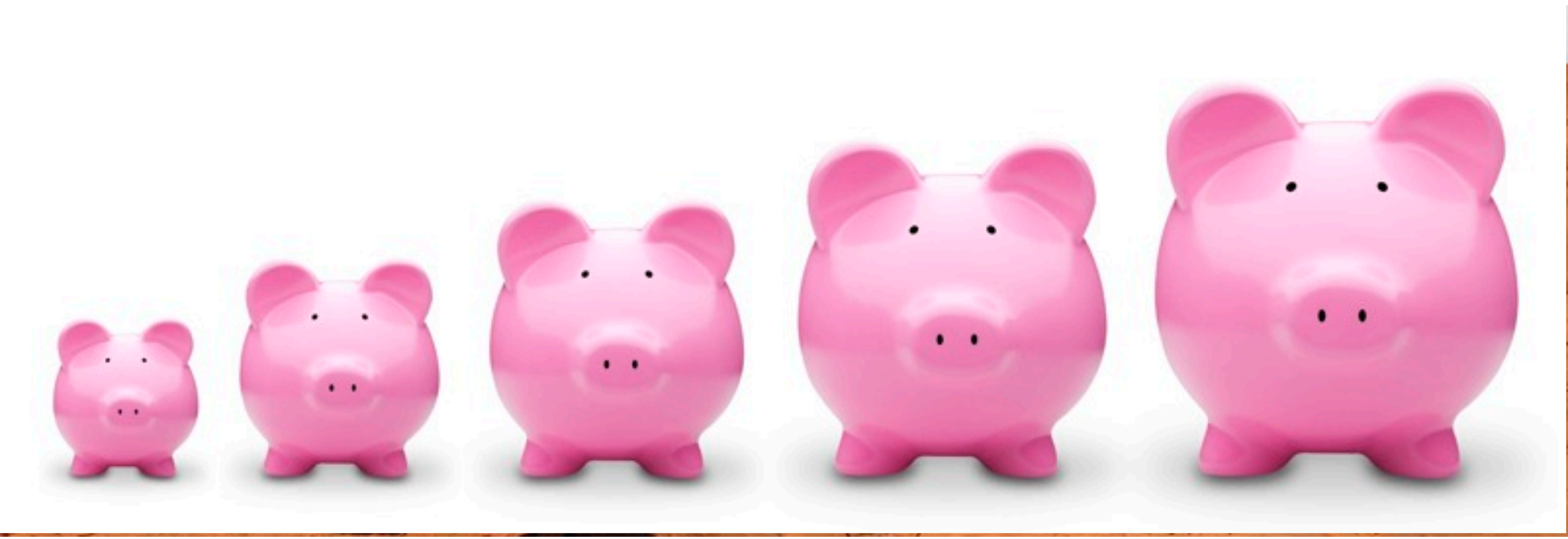
Economic Drivers

- Proportion of electricity in cost of computing
- Product -> Service
- Economies of Scale
- Moore’s Law
- Pay as you go utility model

Cultural Drivers

- Expectations of corporate IT customers have changed
- Consumerization of IT
- Consumer apps more and more like fashion
- Technology achieves ubiquity by disappearing

Access from Anywhere

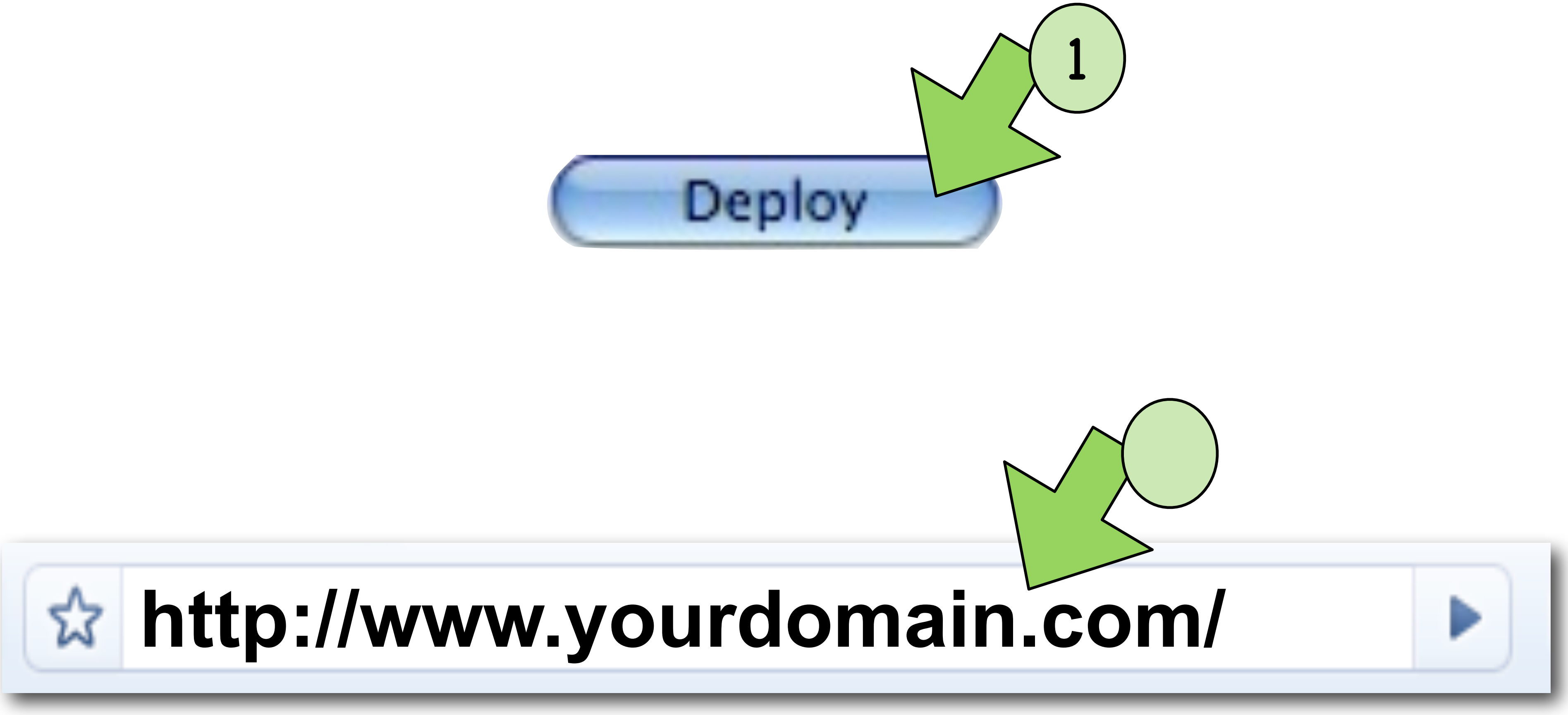


Scales Up, Scales Down, with Demand

Innovation Not Administration

Cultural Drivers: Agility

- Waterfall -> Agile methodologies
- Cloud enables an Agile culture, driver for innovation



http://www.yourdomain.com/

Fail often, fail quickly, and learn

Fail often, fail quickly, and learn

- Risk taking/Experimentation is encouraged
  - http://blog.red-bean.com/sussman/?p=96
- “Do not be afraid of day-to-day failures — learn from them. (As they say at Google, “don’t run from failure — fail often, fail quickly, and learn.”) Cherish your history, both the successes and mistakes. All of these behaviors are the way to get better at programming. If you don’t follow them, you’re cheating your own personal development.”
- Ben Collins-Sussman (Subversion, code.google.com)

Agile Development Processes

Agile Development Processes

- Influences from XP, Agile, Scrum
- Code reviews
- Test Driven Development: Testing on the Toilets program and blog
- Many internal development tools: Mondrian recently open sourced
- Changed the meaning of beta
- Teams co-located: 3-15 people, 4/cubicle, all close to each other
- International offices: manage whole projects, avoid coordination costs

Open Source Culture

Open Source Culture

- Open Source Program Office
- Summer of Code
- Open sourcing parts of Google code
  - http://code.google.com/
- Making the web better: GWT, Gears, OpenSocial, Android

API Culture

API Culture

- Bill Joy: "Innovation happens elsewhere"
- From 3 to 62 APIs in 3 years
- Maps on websites
- Friend Connect: all sites can become social
  - http://code.google.com/ for the list
- Build an ecosystem around the APIs (my job)
- User's choice: get their data out

Data Liberation Front

http://www.dataliberation.org/

Users should be able to control the data they store in any of Google's products. Our team's goal is to make it easier to move data in and out.

Software is moving to the cloud

- What does cloud mean, 4 main angles
  - Delivery 1994 Netscape
  - Infrastructure 2002 Amazon AWS
  - Platform 2008 Google
  - Development now!
- Industrialization of hardware and software infrastructure
  - like electricity beginning of 20th century, cf The Big Switch, Nick Carr
- But software development itself is moving towards a craftmanship

Agility as a survival skill

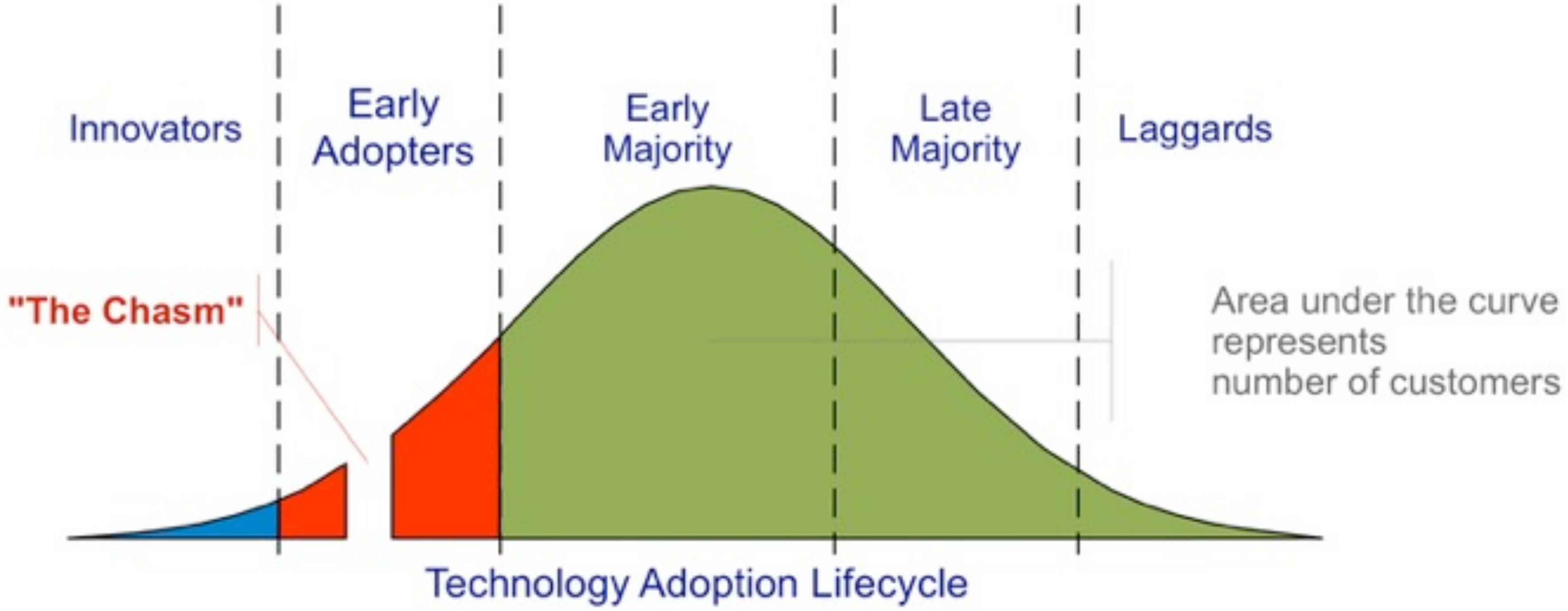
- Software is becoming like fashion
- Phone apps, social apps, short lifetime, fast lifecycles
- Ab testing
- Clay shirky situational apps
- Kent Beck, Usenix 2011 Talk
  change in software process when frequency grows
- Cloud is a powerful driver for agility
- Scalability is built in the platforms
- Can iterate faster
- Focus on design

Chaos of creativity

- Proliferation of languages and frameworks
- Spring, Rails, Grails, Django
- “Pythons has more webframeworks than language keywords”
- Javascript, Python, PHP, Java, Groovy, Scala, Clojure, Go
- Gosling, vm is important, not the language
- Ability to create DSL important, cf Book
- Fragmentation of communities
- Chaotic Darwinian period, fun for the curious, deadly for the ossified
- Online services replacing a lot of software
- Mashups, Weaving services together
- Pick your battles, choose what you need to build yourself to add value

Crossing the Chasm

- Build the whole product
- Cloud getting mainstream: Apple iCloud
- Opportunities and risks
- Ecosystems, various platforms


Innovators
Early Adopters
Early Majority
Late Majority
Laggards
"The Chasm"
Area under the curve represents number of customers
Technology Adoption Lifecycle

Picture from Wikimedia Foundation http://en.wikipedia.org/wiki/File:Technology-Adoption-Lifecycle.png

Delivery/Monetization/Marketing

- Appstores, saas, social media
- Opportunities, story kieden
- Risks, fragmentation, multiplicity, lack of cross platform
- Be your own bitch, understand platform strategies, leverage and not be
- used, story tweetdeck vs seesmic

Infrastructure

- Aws, joyent, rackspace
- Start of standardization
- Depends on size, economies of scale
- Be your own bitch, build distributed platform on top of infrastructure
- Story aws meltdown[b]
- http://blog.reddit.com/2011/03/why-reddit-was-down-for-6-of-last-24.html
- http://www.readwriteweb.com/cloud/2010/12/chaos-monkey-how-netflix-uses.php
- http://news.ycombinator.com/item?id=2477296
- http://stu.mp/2011/04/the-cloud-is-not-a-silver-bullet.html
- twilio, smugmug, simplegeo survived

Be your own bitch

“Don’t be a Google Bitch, don’t be a Facebook Bitch, and Don’t be a Twitter Bitch. Be your own Bitch.”

Fred Wilson

http://techcrunch.com/2011/05/23/fred-wilson-be-your-own-bitch/

Future of Infrastructure

- Future: consider Infrastructure as CDNs today, multi cloud usage
- Issue, replication, bandwidth
- Open source, open standards, deltacloud, openstack, eucalyptus
- A lot of fighting in is area this year
- Be your own bitch: use openstack or deltacloud and use several providers

Platforms

- Web stack, nosql, sql
- Google App Engine, Joyent, Heroku, Stax (Cloudbees), Amazon elastic beanstalk, Microsoft Azure
- Single or a few languages, services
- Start multi language platforms, dotcloud
- Lack of standards: risk, vendor lock-in

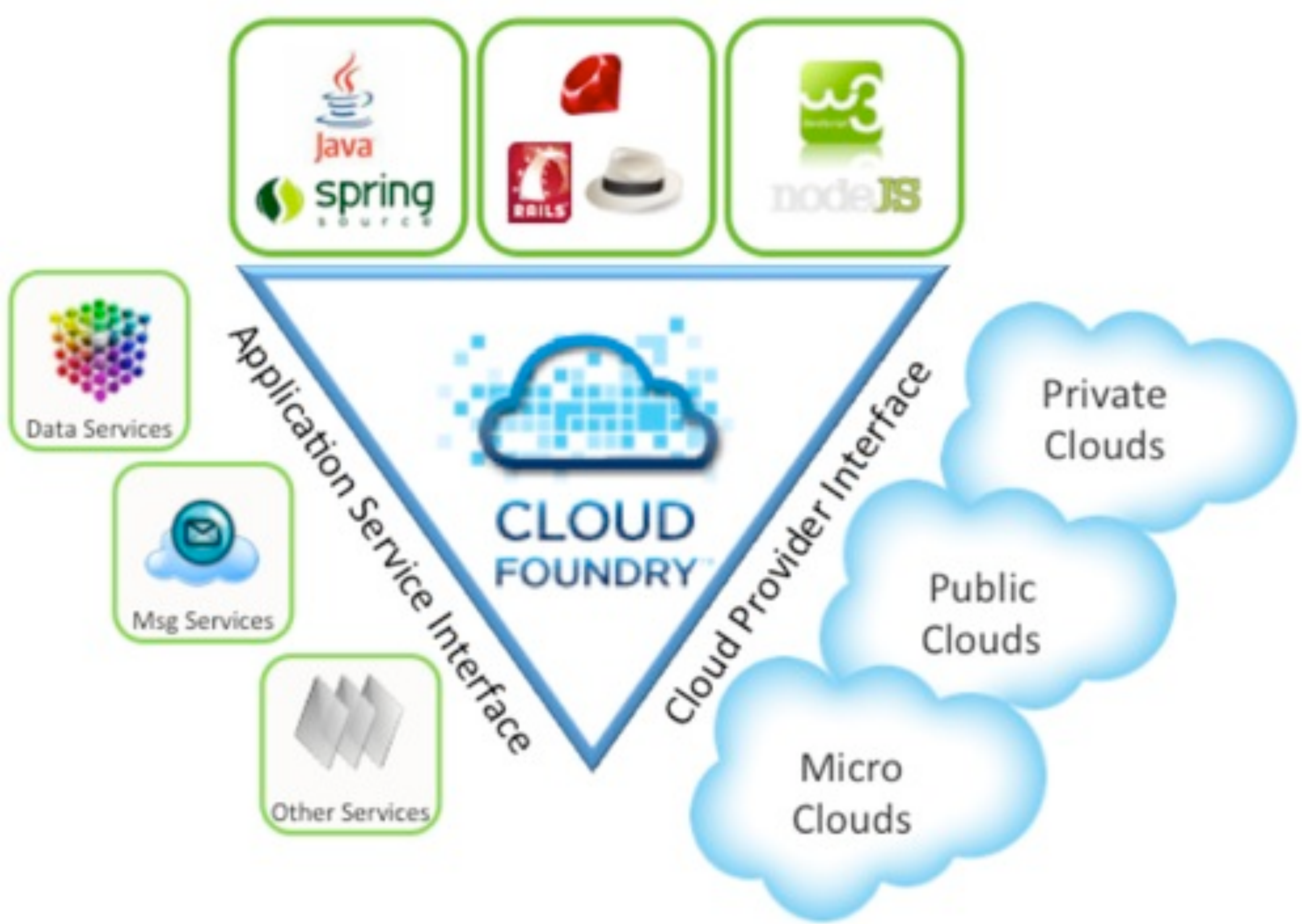
Main Risk: Lock-In

Welcome to the hotel california
Such a lovely place
Such a lovely face
Plenty of room at the hotel california
Any time of year, you can find it here

Last thing I remember, I was
Running for the door
I had to find the passage back
To the place I was before
'relax,' said the night man,
We are programmed to receive.
You can checkout any time you like,
But you can never leave!

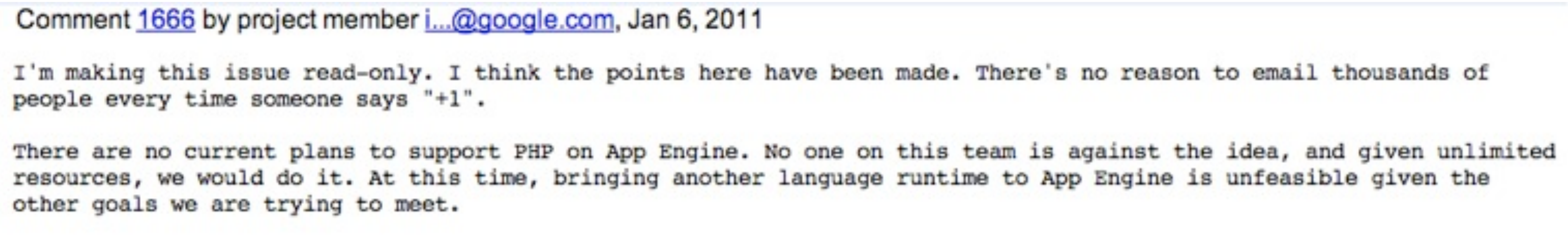
Cloud Foundry

- Be your own bitch, today Cloud Foundry - Apache 2 Licensed
  - multi language/frameworks
  - multi services
  - multi cloud



Open Source Advantage

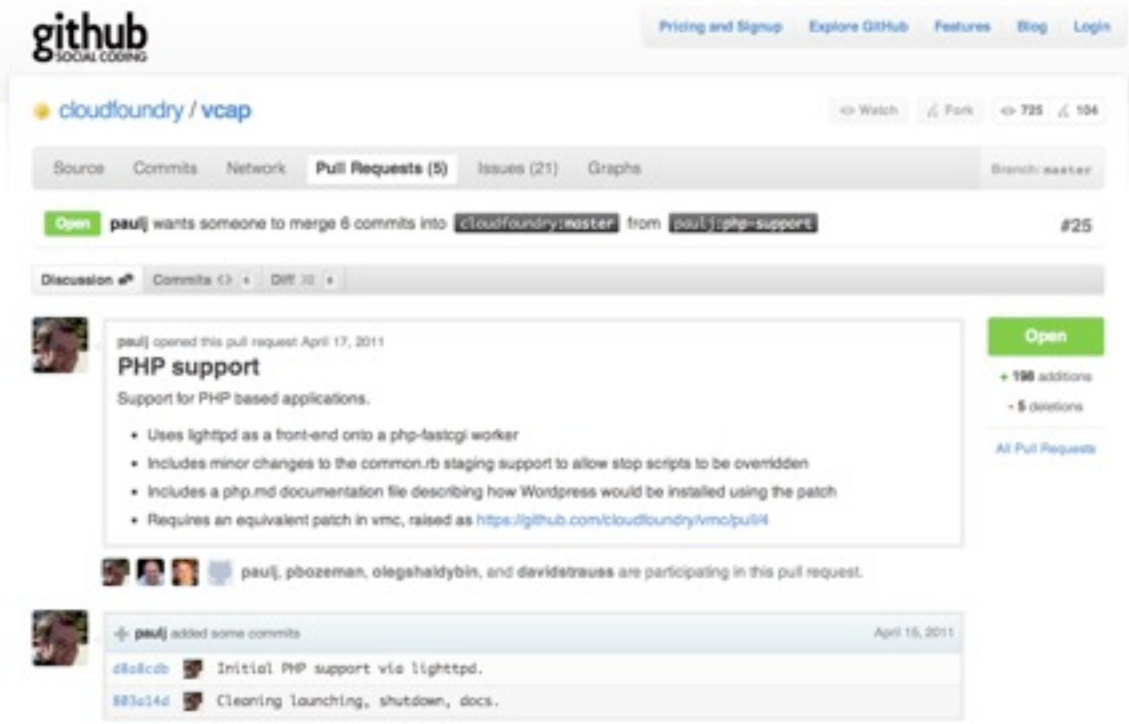
- http://code.google.com/p/googleappengine/issues/detail?id=13



Comment 1666 by project member i...@google.com, Jan 6, 2011

```
I'm making this issue read-only. I think the points here have been made. There's no reason to email thousands of
people every time someone says "+1".

There are no current plans to support PHP on App Engine. No one on this team is against the idea, and given unlimited
resources, we would do it. At this time, bringing another language runtime to App Engine is unfeasible given the
other goals we are trying to meet.
```

- https://github.com/cloudfoundry/vcap/pull/25



Open paulj wants someone to merge 6 commits into cloudfoundry:master from paulj:php-support #25

paulj opened this pull request April 17, 2011

PHP support

Support for PHP based applications.

- Uses lighttpd as a front-end onto a php-fastcgi worker
- Includes minor changes to the common.rb staging support to allow stop scripts to be overridden
- Includes a php.md documentation file describing how Wordpress would be installed using the patch
- Requires an equivalent patch in vmc, raised as https://github.com/cloudfoundry/vmc/pull/4

+ 198 additions

- 5 deletions

paulj, pbozeman, olegshaldybin, and davidstrauss are participating in this pull request.

paulj added some commits April 15, 2011

d8c8cdb Initial PHP support via lighttpd.

803c14d Cleaning launching, shutdown, docs.

BigData Platforms: Hadoop

- Apache Hadoop, open source version of Google MapReduce, GFS...
- Cloudera, many others, space heating up
- EMC, HortonWorks distros
- Google Bigquery
- Be your own bitch, today, Cloudera distro

Services

- Services
- Apis, apigee, mashery
- Telephony, Twilio
- Geo
- Social
- Visualization

Development

- Final fronteer, happening now
- Not whole product yet
- Scm, dev, build, test, prod, community
- Scm, google code, github
- Dev cloud9, orion, exo
- Higher level case tools, wavemaker, orangescape, runmyprocess
- Build Cloudbees, dev and prod clouds
- Story didier girard
- Test, feature of cloud platforms
- Community stackoverflow, quora?, startup doing code analysis

Reinventing yourself

- Things to forget
  - First normal form, waterfall model, single server development, single
  - language skills
- Things to learn and embrace
  - Agile, api design, Ui design, javacript, html5, css3, ab testing, open
  - source, open standards, architecture, distributed computing (caps
  - theorem, 8 fallacies) cloud platforms and api, multiple types of
  - languages (imperative, object, functional, logic), reading T&Cs
  - Learn to live in a box (embrace platform limitations) to think outside the box

Predictions

- Software is becoming like fashion, design rules
- Welcome to Babel, use the best tool for the job, embrace multiple language & heterogeneity
- Our jobs will change, build yourself out of your current job
- Sysadmin jobs will disappear, except at large cloud providers
- Many opportunities open when you embrace change

What it means for you

- Build On the shoulders of giants
- Take risks, to innovate, story ebay
- Learn everyday, try different things
- learn an api / month, a language / year
- Be fast and agile
- Make money
- Social and app stores

What it means for you

- Be your own bitch
- Look at open source / open standards aspects of the platforms and services you use
- Like a kid on a candy store, there's never been a better time to be a software developer
- Welcome to the Cloud, embrace change and reinvent yourselves
- “The future is already there, not evenly distributed” Gibson
- We Developers, invent the future today

Books / Articles

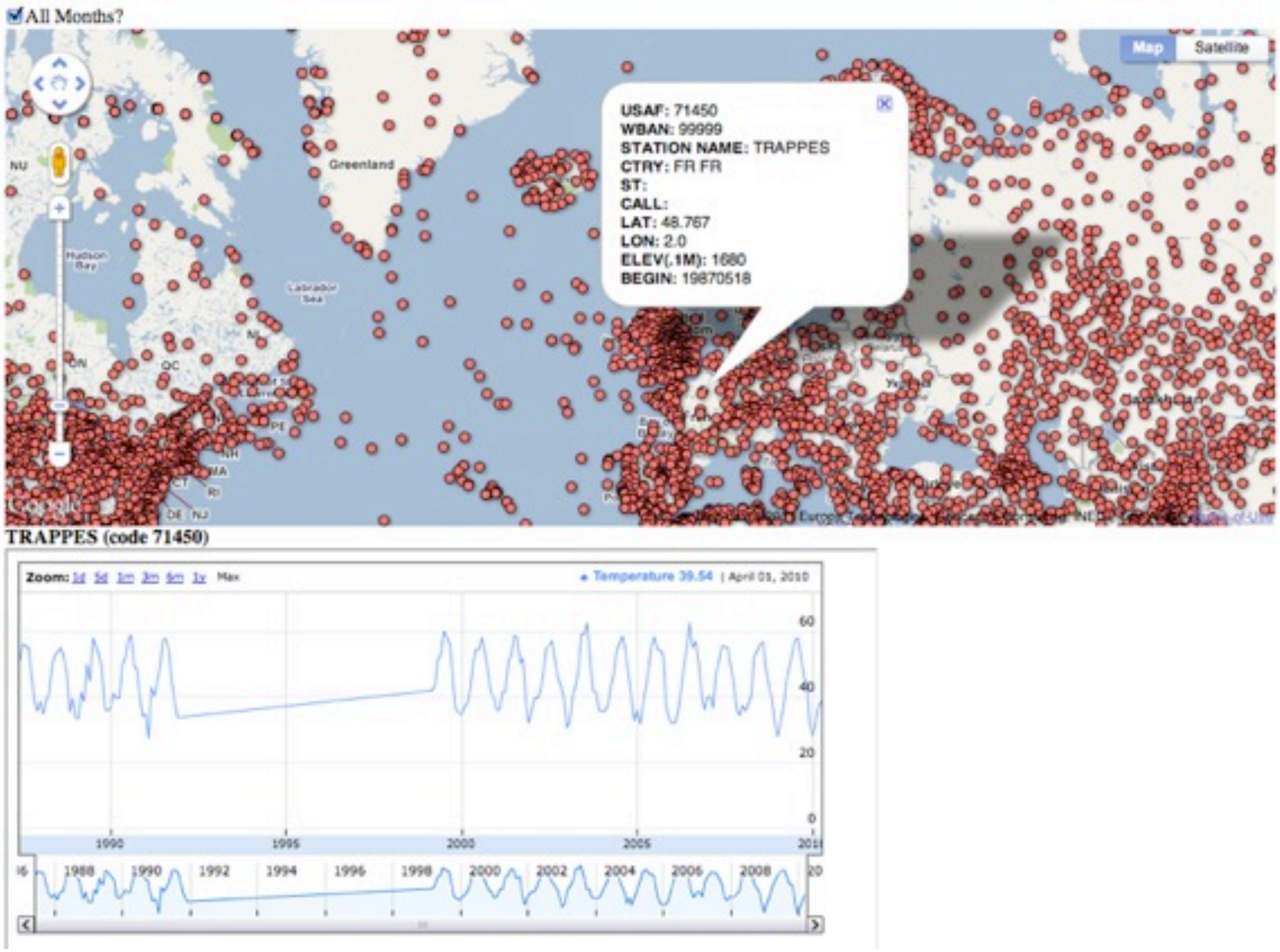
- Nick Carr, The Big Switch
- Eric Raymond, The Art of Unix Programming
- Weinberg, Psychology of Computer Programming
- Wes python book
- Mark html5 book
- Kent Beck XP
- Hunt, Thomas, The Pragmatic Programmer
- Ade Oshineye, Apprenticeship Patterns
- Matt Cutt's Ignite Talk IO 2011, Trying different things
- Josh Bloch talk about api design
- Larry and Sergey, Anatomy of a Search Engine
- Rob Pike, The Practice of Programming

Papers / Talks

- Simon Wardley, Oscon 09 “Cloud - Why IT Matters”
- Tim O’Reilly article on internet os
- Peter Deutsch’s 8 Fallacies of Distributed Computing
- Brewer’s CAP Theorem
- Gregor Hohpe’s Starbucks Does Not Use Two-Phase Commit
- Stuff I tag http://www.delicious.com/chanezon/
- My previous Talks http://www.slideshare.net/chanezon
- My list of favorite books http://www.chanezon.com/pat/soft_books.html

Demo: Historical Weather Data Browsing

- App Engine, Fusion Tables, BigQuery, Visualization API



☑ All Months?

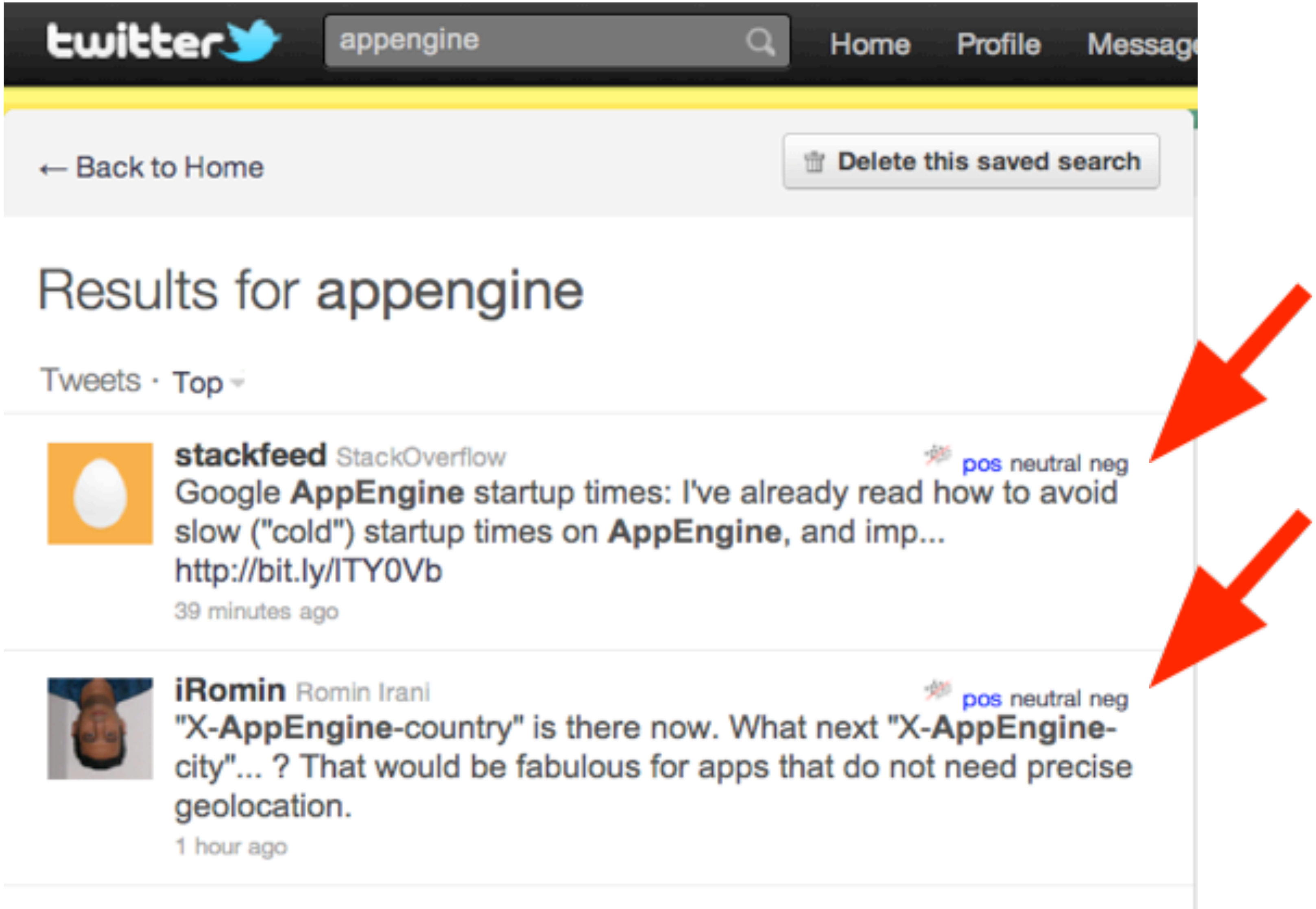
USAF: 71450
WBAN: 99999
STATION NAME: TRAPPES
CTRY: FR FR
ST:
CALL:
LAT: 48.767
LON: 2.0
ELEV(.1M): 1680
BEGIN: 19870518

TRAPPES (code 71450)

Temperature 39.54 | April 01, 2010

Demo Tweet Sentiment Analysis

- App Engine, Google Storage, Google Prediction, Chrome Extension
- Nick Johnson, Wesley Chun, Patrick Chanezon



Results for appengine

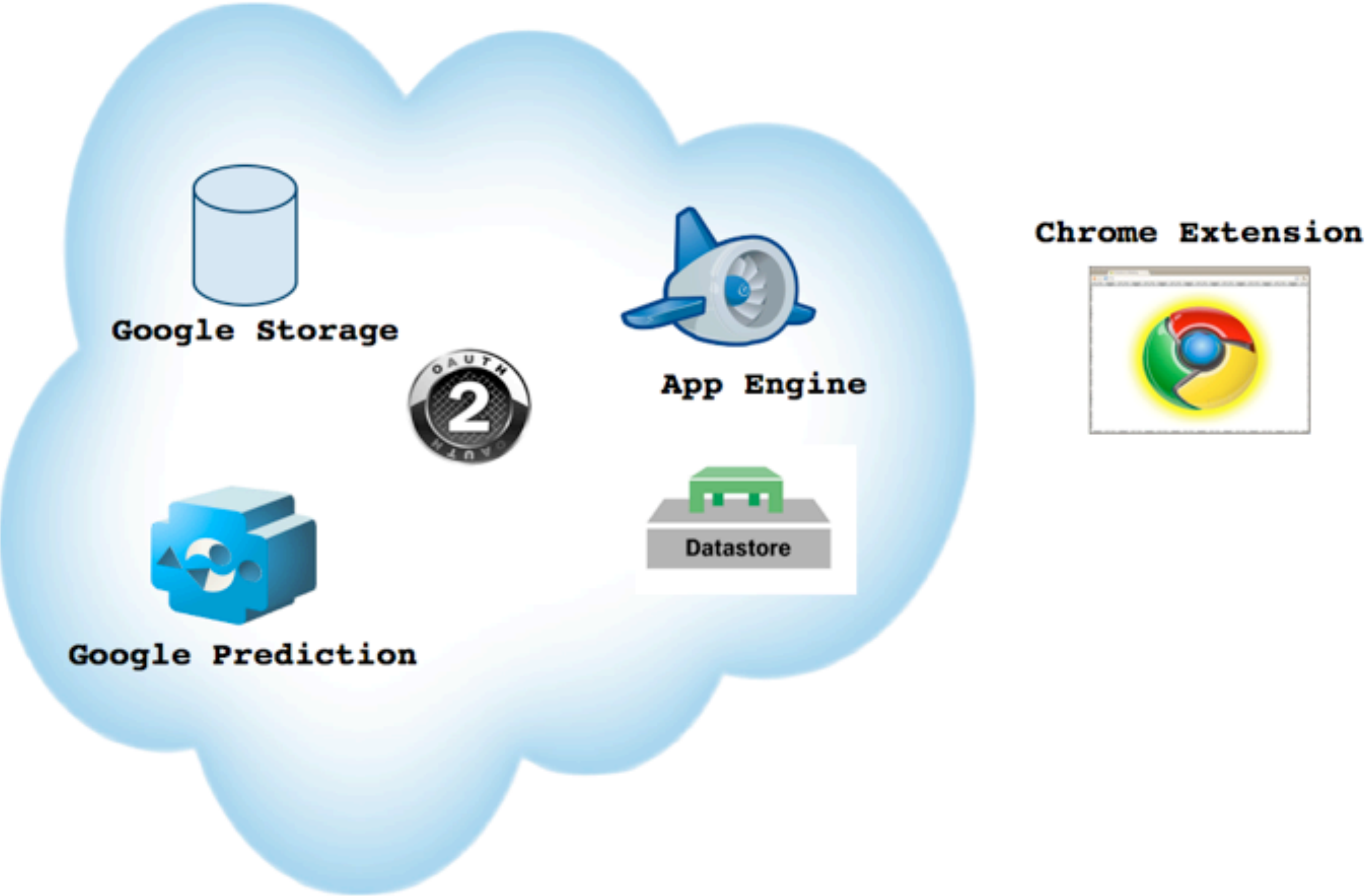
stackfeed StackOverflow
Google **AppEngine** startup times: I've already read how to avoid slow ("cold") startup times on **AppEngine**, and imp... http://bit.ly/lTY0Vb
39 minutes ago

iRomin Romin Irani
"X-**AppEngine**-country" is there now. What next "X-**AppEngine**-city"... ? That would be fabulous for apps that do not need precise geolocation.
1 hour ago

Tweet Sentiment Analysis

- Let users create models to predict Tweet categories
  - Tweets are categorized directly from the Tweeter UI using a Chrome extension
  - Access Control: teams can create and manage models
  - Tweets + categories are stored in Bigtable, then sent to Storage to create a Prediction API model
  - The models can be used by the extension to autocategorize Tweets the user sees
  - Or they can be used offline by the App to create daily dashboards
  - Initial version created during the Cloud hackathon in April
  - Uses Chrome Extension, App Engine, Storage, Prediction
  - Leveraged Seth Ladd's +1 Chrome Extension sample

Tweet Sentiment Analysis: Architecture



Google Storage

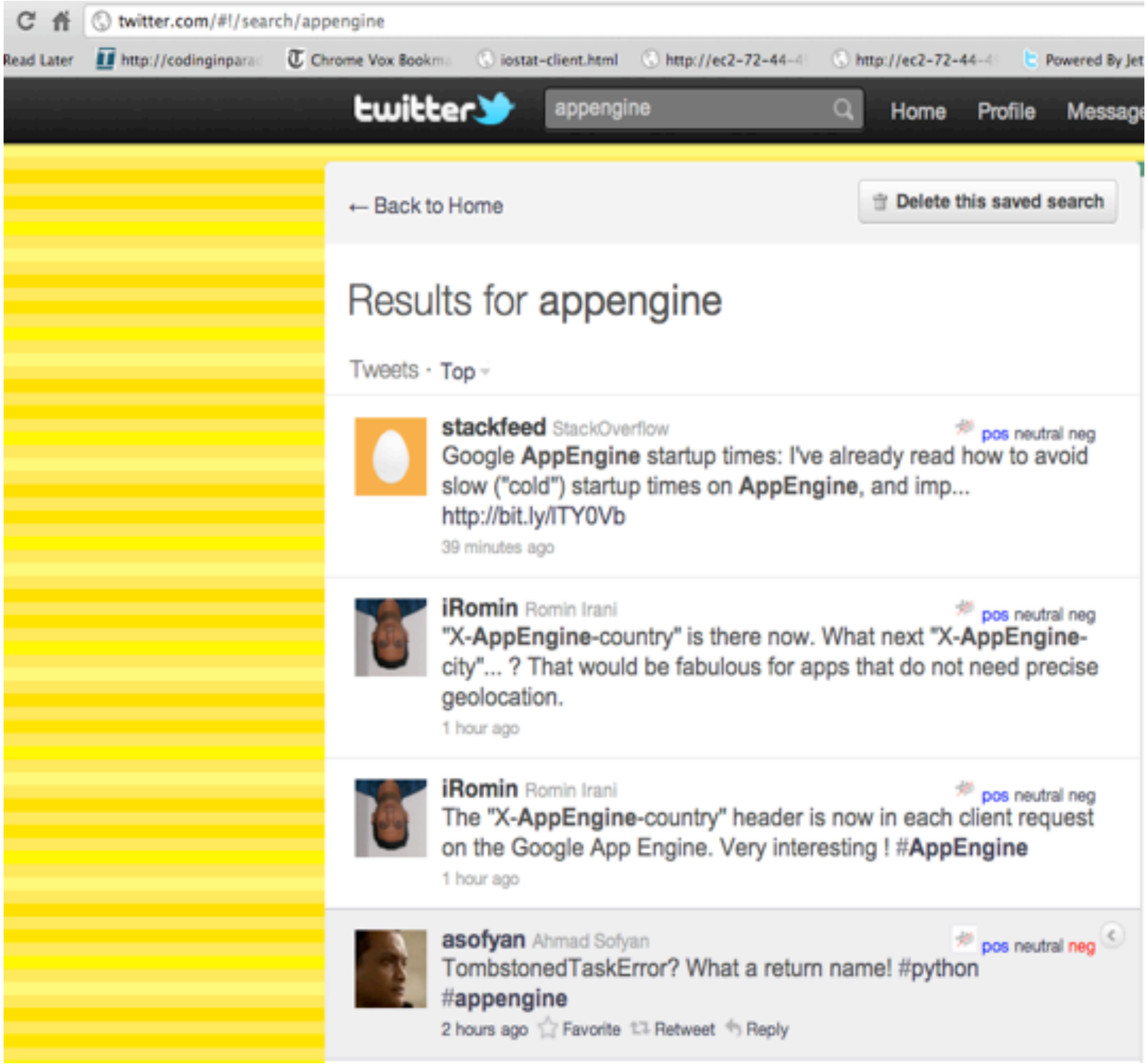
App Engine

Datastore

Google Prediction

Chrome Extension

Tweet Sentiment Analysis: Demo



Results for appengine

Tweets · Top

stackfeed StackOverflow — pos neutral neg
Google **AppEngine** startup times: I've already read how to avoid slow ("cold") startup times on **AppEngine**, and imp... http://bit.ly/lTY0Vb
39 minutes ago

iRomin Romin Irani — pos neutral neg
"X-**AppEngine**-country" is there now. What next "X-**AppEngine**-city"... ? That would be fabulous for apps that do not need precise geolocation.
1 hour ago

iRomin Romin Irani — pos neutral neg
The "X-**AppEngine**-country" header is now in each client request on the Google App Engine. Very interesting ! #**AppEngine**
1 hour ago

asofyan Ahmad Sofyan — pos neutral neg
TombstonedTaskError? What a return name! #python #**appengine**
2 hours ago ☆ Favorite Retweet Reply

Tweet Sentiment Analysis: Status

- Release Plan
  - Code at http://code.google.com/p/gae-tweet-sentiment-analysis/
  - Demo at http://pat-social.appspot.com/
  - Should have a finalized usable version end of summer

Q&A

Didier Girard, Sfeir

- Cloud pour une SSII
- Cloud et Agilite
- Cloudbees, App Engine

Erwan Arzur, RunMyProcess

- Cloud for an ISV
- AWS
- Google Apps MarketPlace

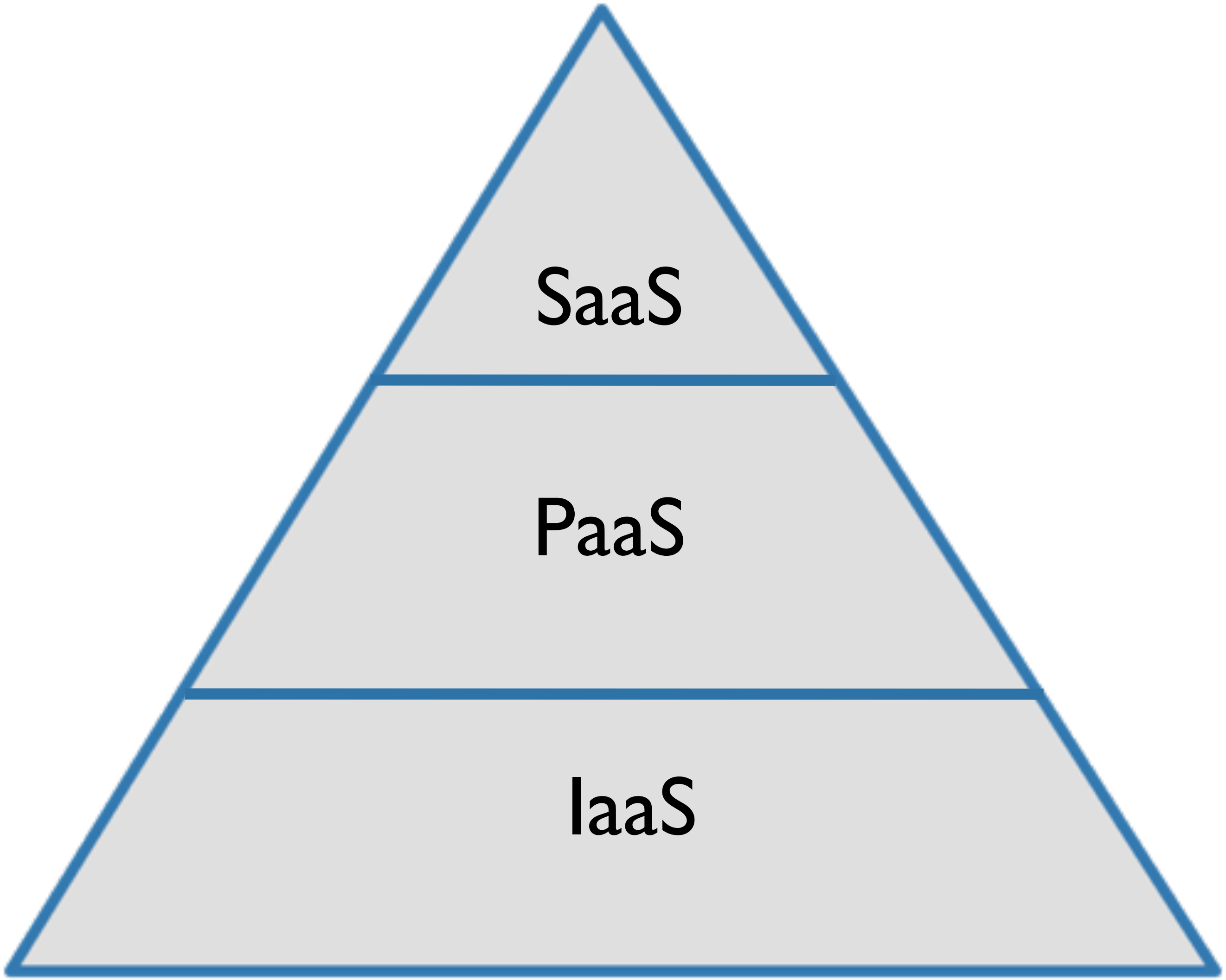
Guillaume Laforge, VMWare/SpringSource

- Cloud Foundry, an Open Source Cloud Platform
- Groovy in the Cloud

Jeremi Joslin, Exo Platforms

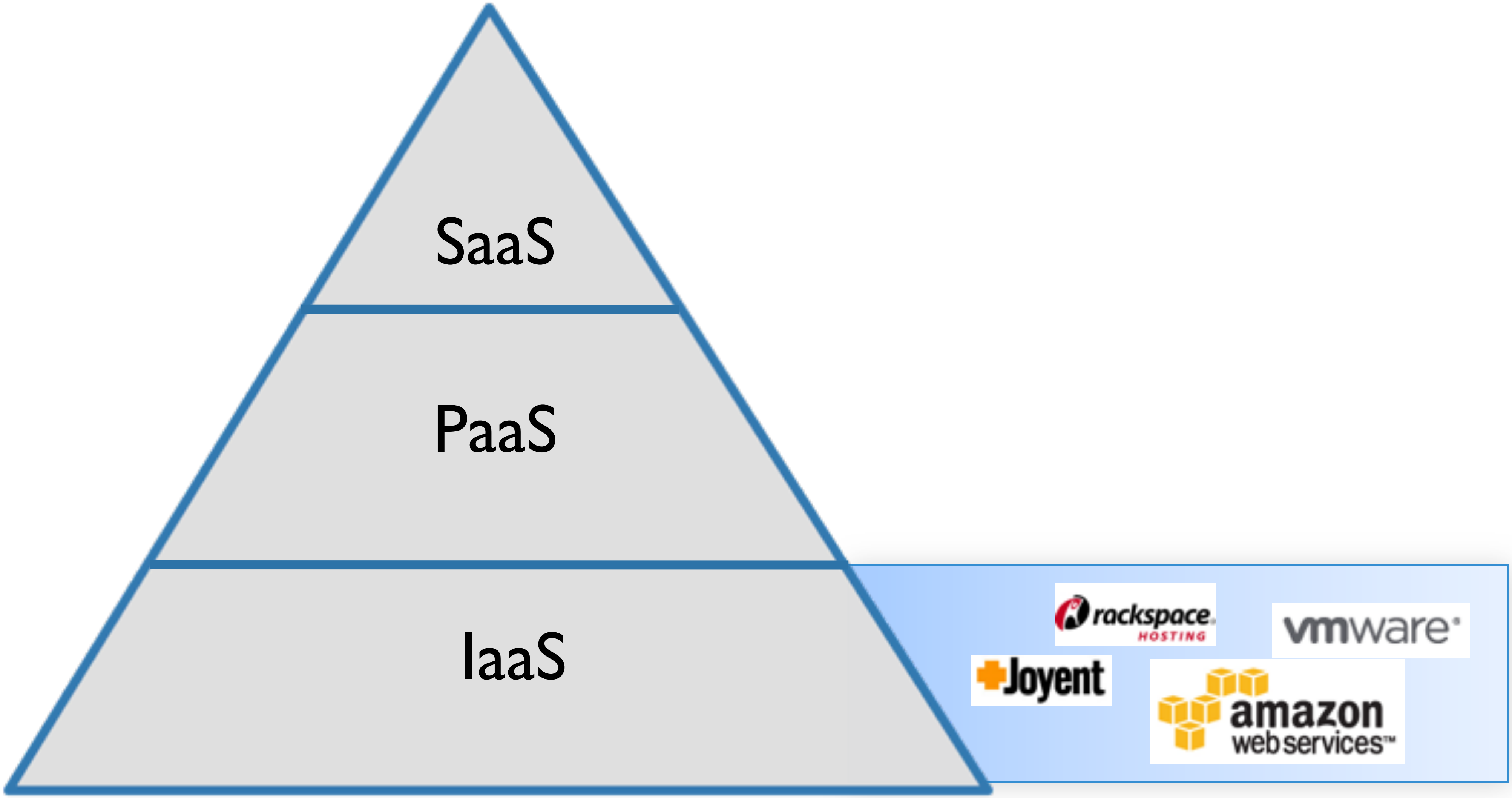
- Cloud IDE,
- demo of Exo Cloud IDE

Cloud Computing Categories



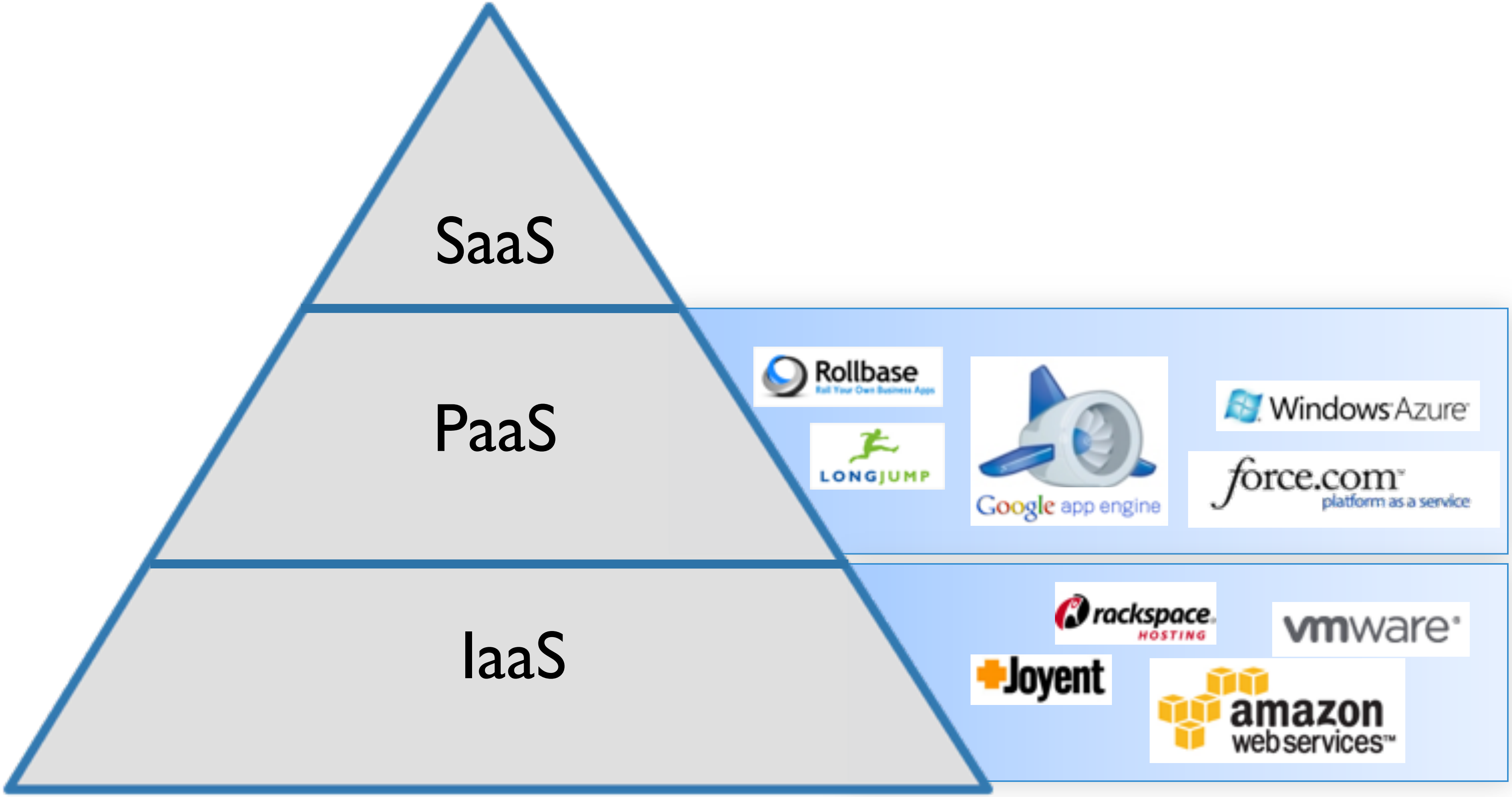
Source: Gartner AADI Summit Dec 2009

Cloud Computing Categories



Source: Gartner AADI Summit Dec 2009

Cloud Computing Categories



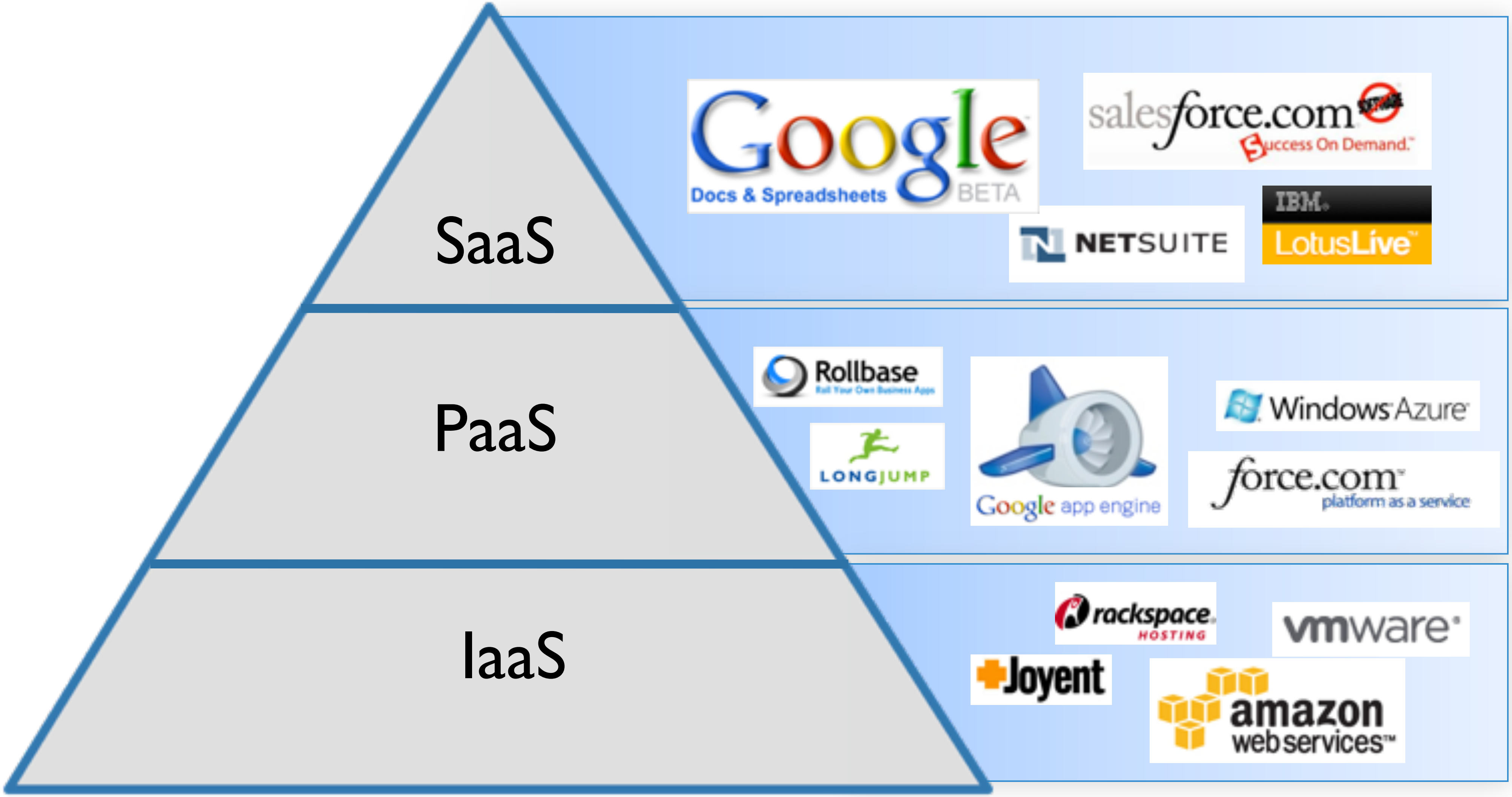
SaaS

PaaS

IaaS

Source: Gartner AADI Summit Dec 2009

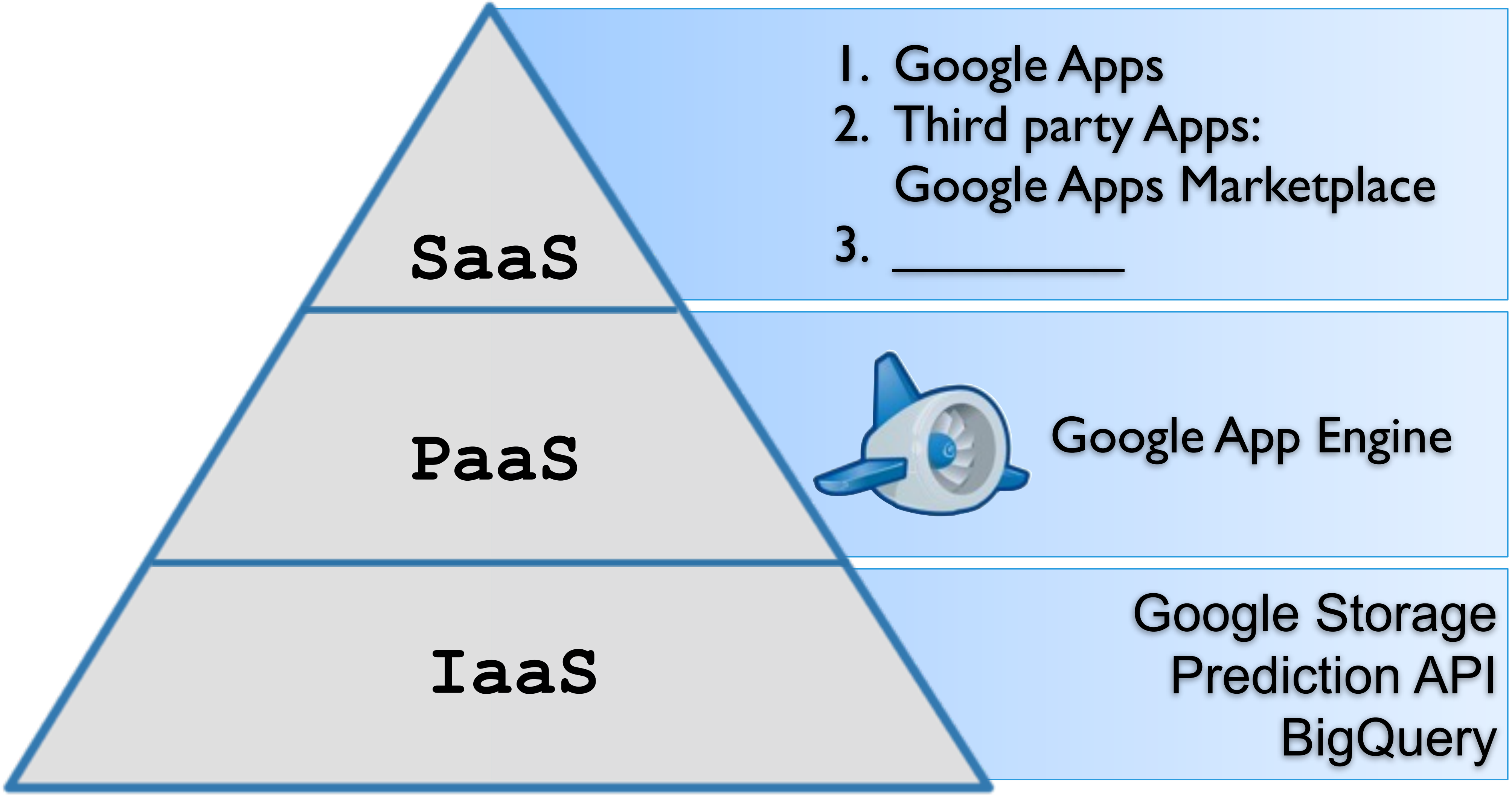
Cloud Computing Categories


SaaS
Google Docs & Spreadsheets BETA
salesforce.com Success On Demand.
NETSUITE
IBM. LotusLive
PaaS
Rollbase Roll Your Own Business Apps
LONGJUMP
Google app engine
Windows Azure
force.com platform as a service
IaaS
rackspace HOSTING
vmware
Joyent
amazon web services

Source: Gartner AADI Summit Dec 2009

Google Cloud Products

Google's Cloud Offerings



SaaS

1. Google Apps
2. Third party Apps:
   Google Apps Marketplace
3. ____________

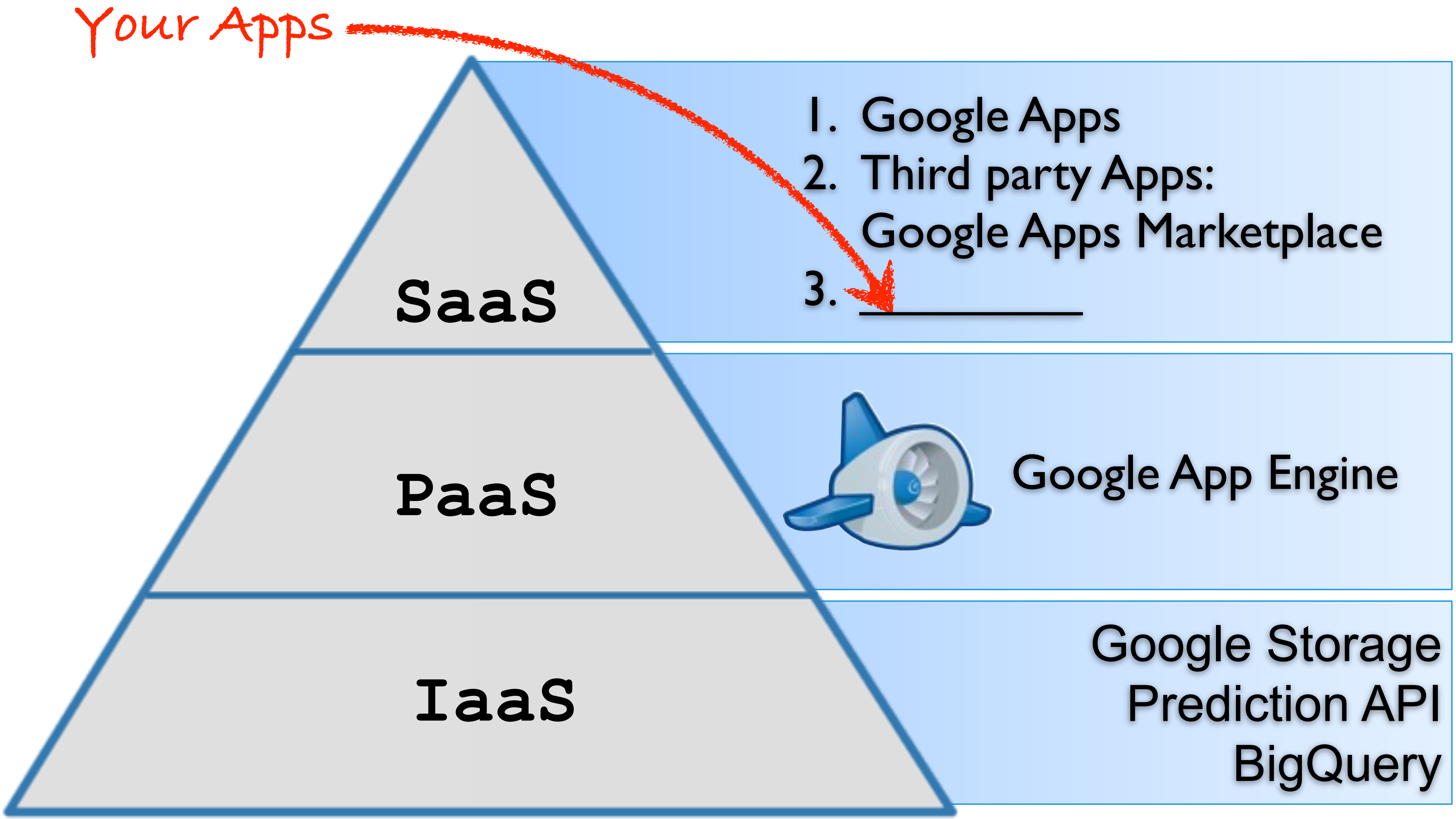
PaaS

Google App Engine

IaaS

Google Storage
Prediction API
BigQuery

Google's Cloud Offerings



Your Apps

SaaS

1. Google Apps
2. Third party Apps:
   Google Apps Marketplace
3. ________

PaaS

Google App Engine

IaaS

Google Storage
Prediction API
BigQuery

How Google Apps Adds Value

Productivity and Innovation

Realtime collaboration, constant updates, new features

Platform Independence

Work anywhere from any computer or mobile device

Reduced IT Complexity

Least complex, least expensive to license and manage

How Google Apps Adds Value

Security and Availability

Same uptime and infrastructure used for Google products

Built-in Enterprise Security Features

2-Factor Authentication, Single Sign On, Reporting Tools

How Google Apps Adds Value

Security and Availability

Same uptime and infrastructure used for Google products

Built-in Enterprise Security Features

2-Factor Authentication, Single Sign On, Reporting Tools

A Toolbox of Administrative APIs

Reporting, Compliance, Identity Management and more...

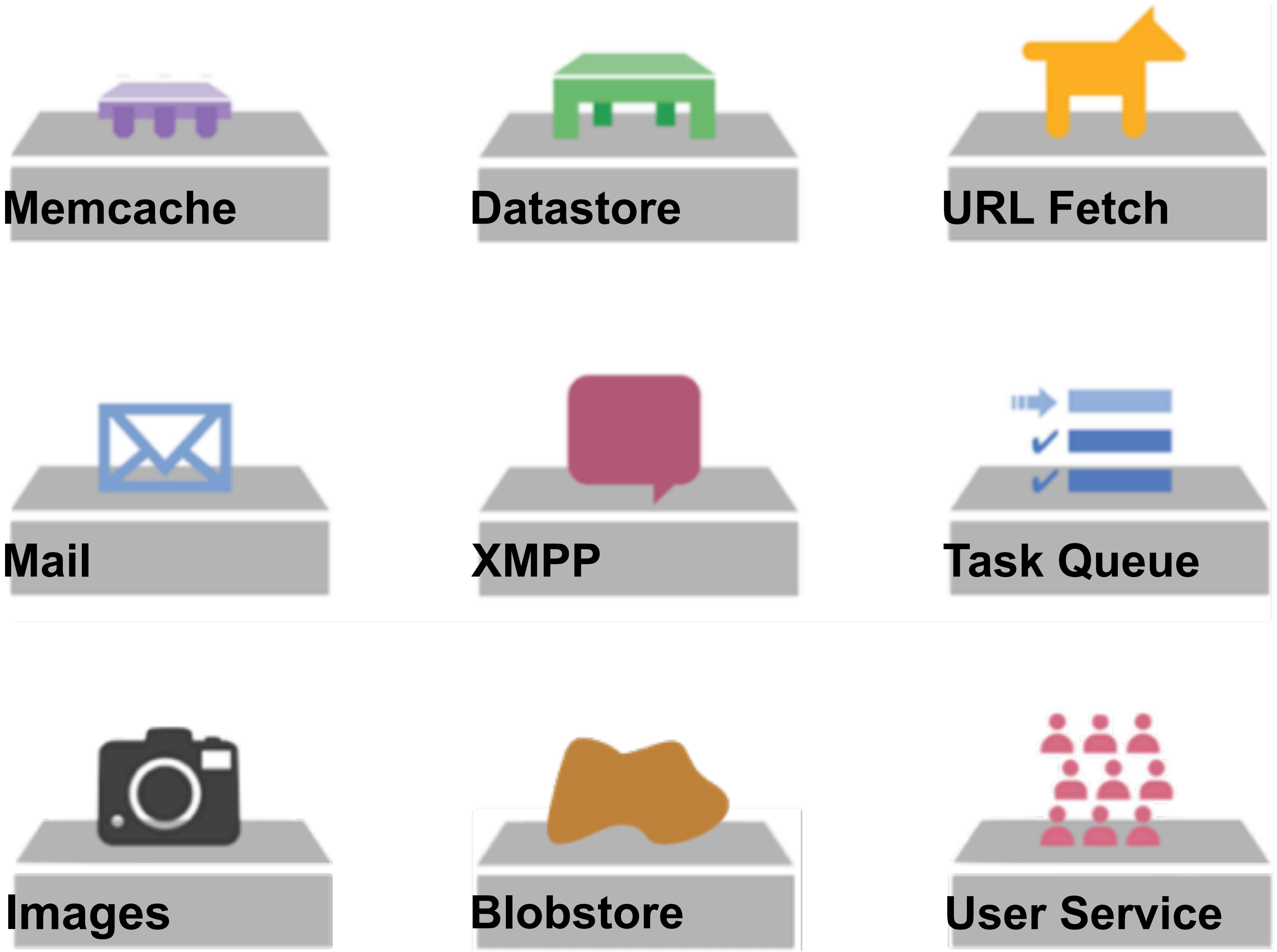
Why Google App Engine?

- Easy to build
- Easy to maintain
- Easy to scale

Cloud Development in a Box

- Downloadable SDK
- Application runtimes
  - Java, Python
- Local development tools
  - Eclipse plugin, AppEngine Launcher
- Specialized application services
- Cloud based dashboard
- Ready to scale
- Built in fault tolerance, load balancing

Specialized Services



Memcache

Datastore

URL Fetch

Mail

XMPP

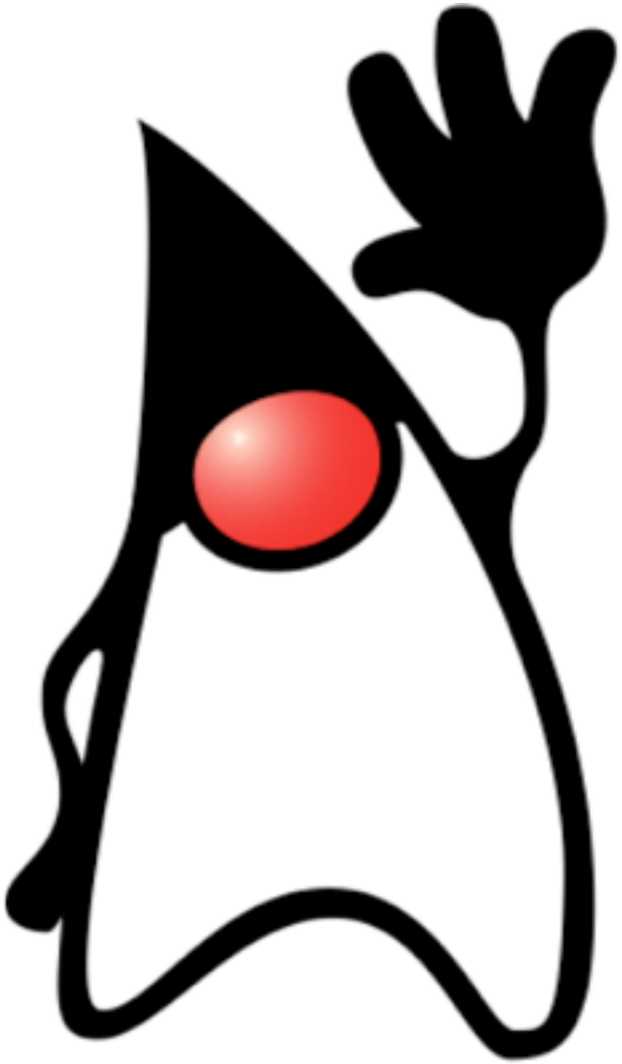
Task Queue

Images

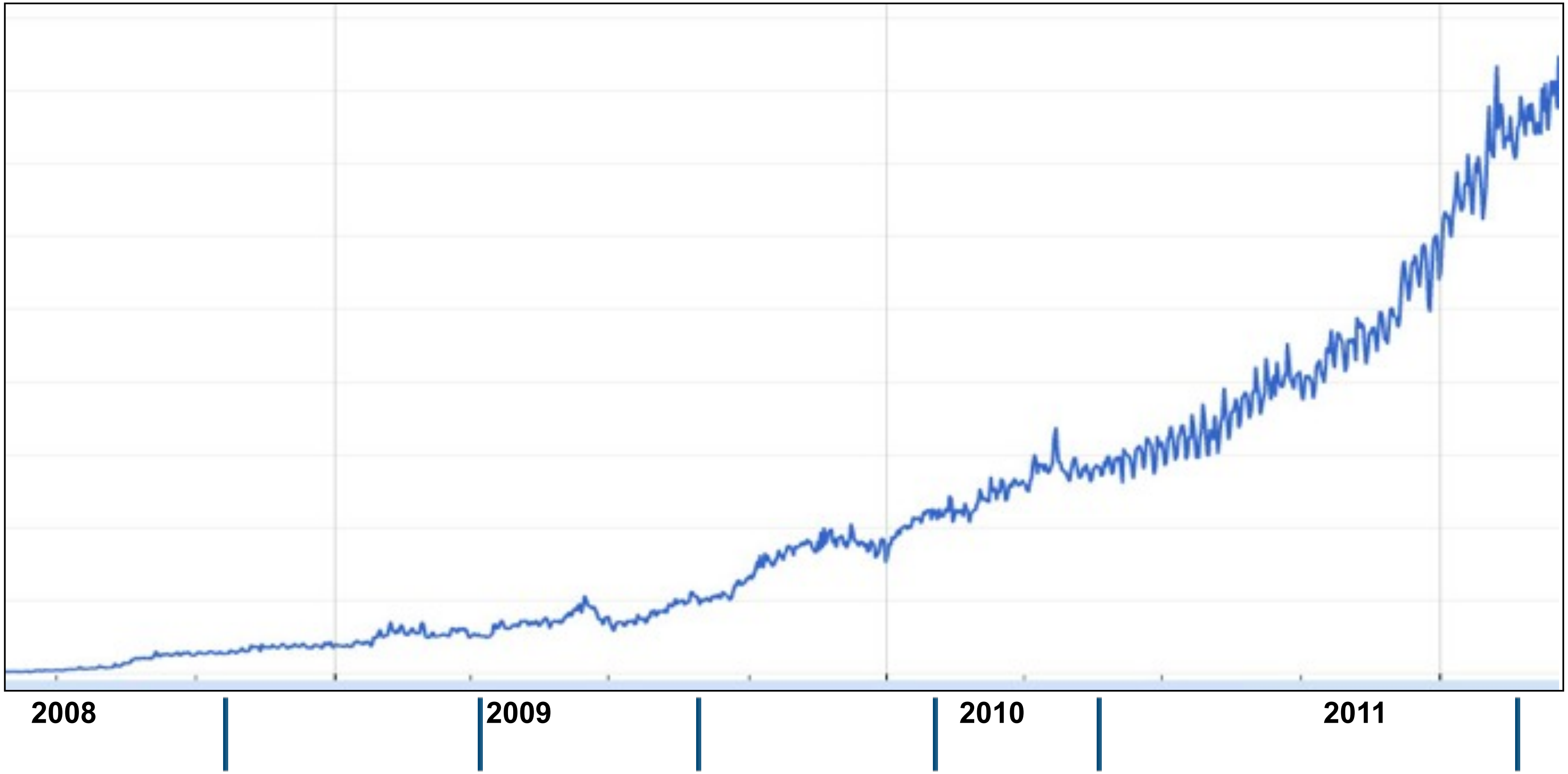
Blobstore

User Service

Language Runtimes



App Engine Growth



App Engine Launch
Python
Datastore
Memcache
logs export

Batch write/read
Https
Status-
Dashboard

Java
DB Import
cron

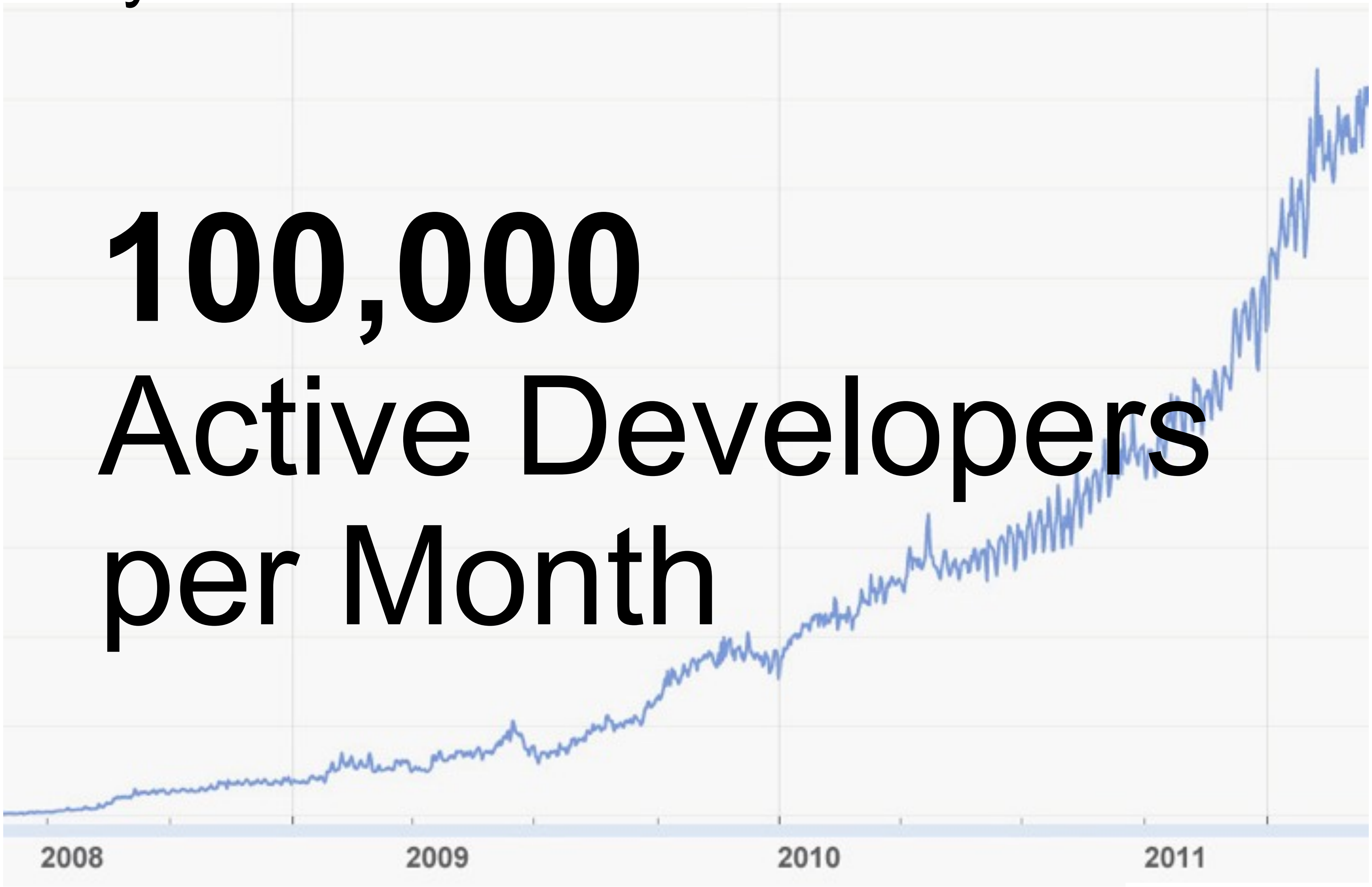
Task Queues
XMPP
incoming email

Blobstore
Appstats
cursors
Mapper

Multitenancy
Instance Console
Always On
hi-perf imag
10 min tasks

Hi-Replication
Datastore
Channel API
Files API
Remote API
Prosp Search

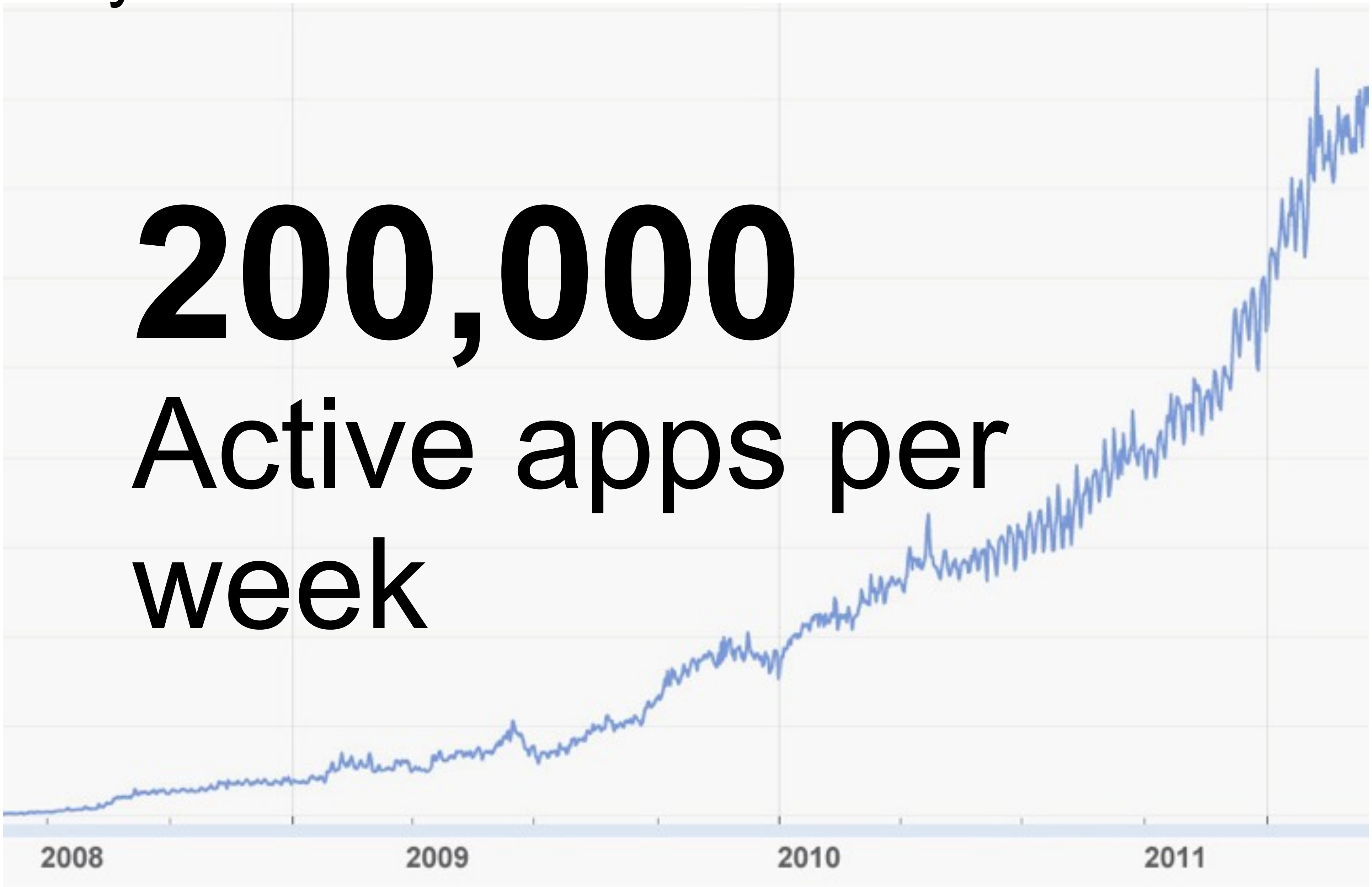
By the Numbers



100,000

Active Developers per Month

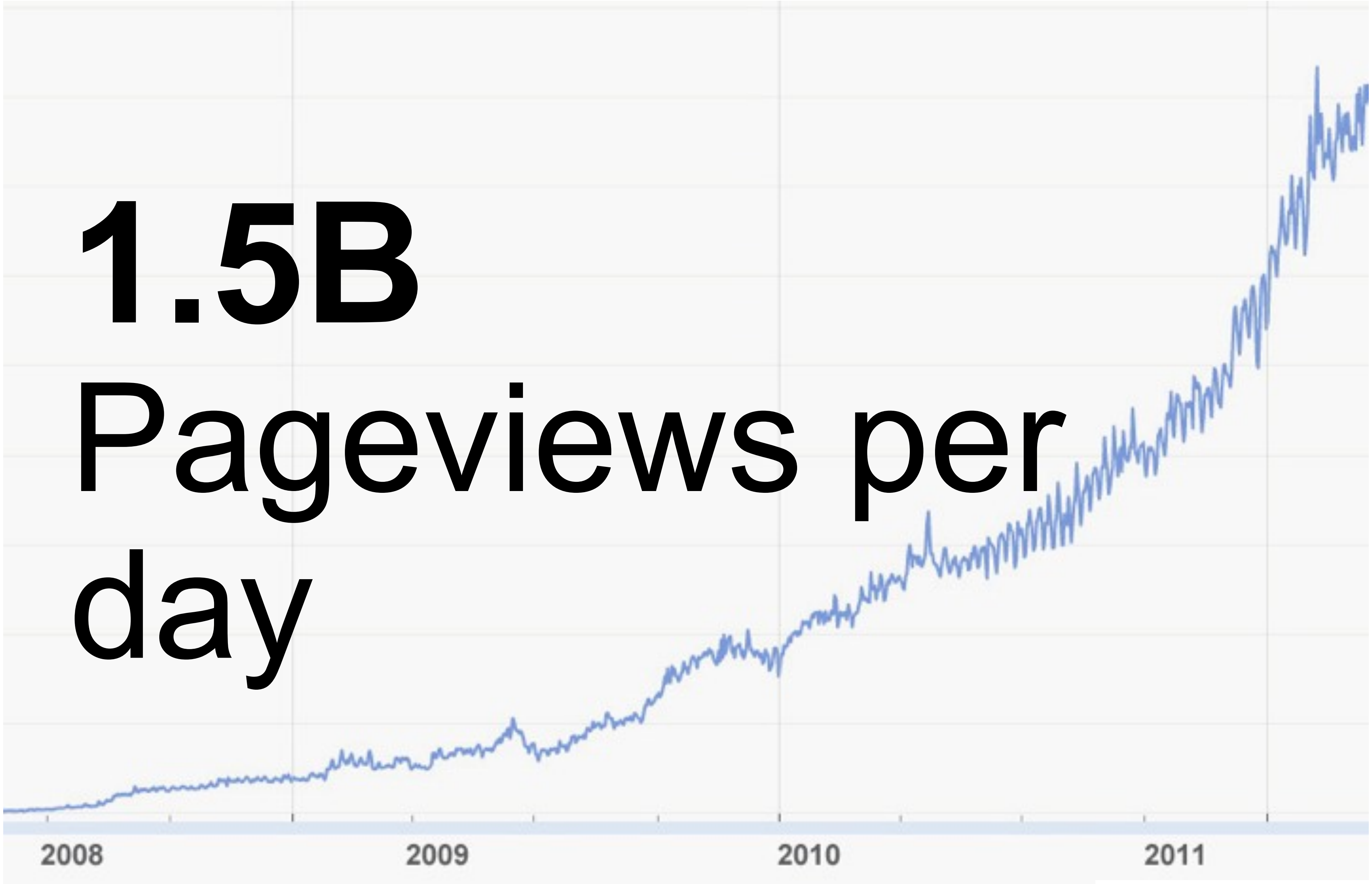
By the Numbers



200,000

Active apps per week

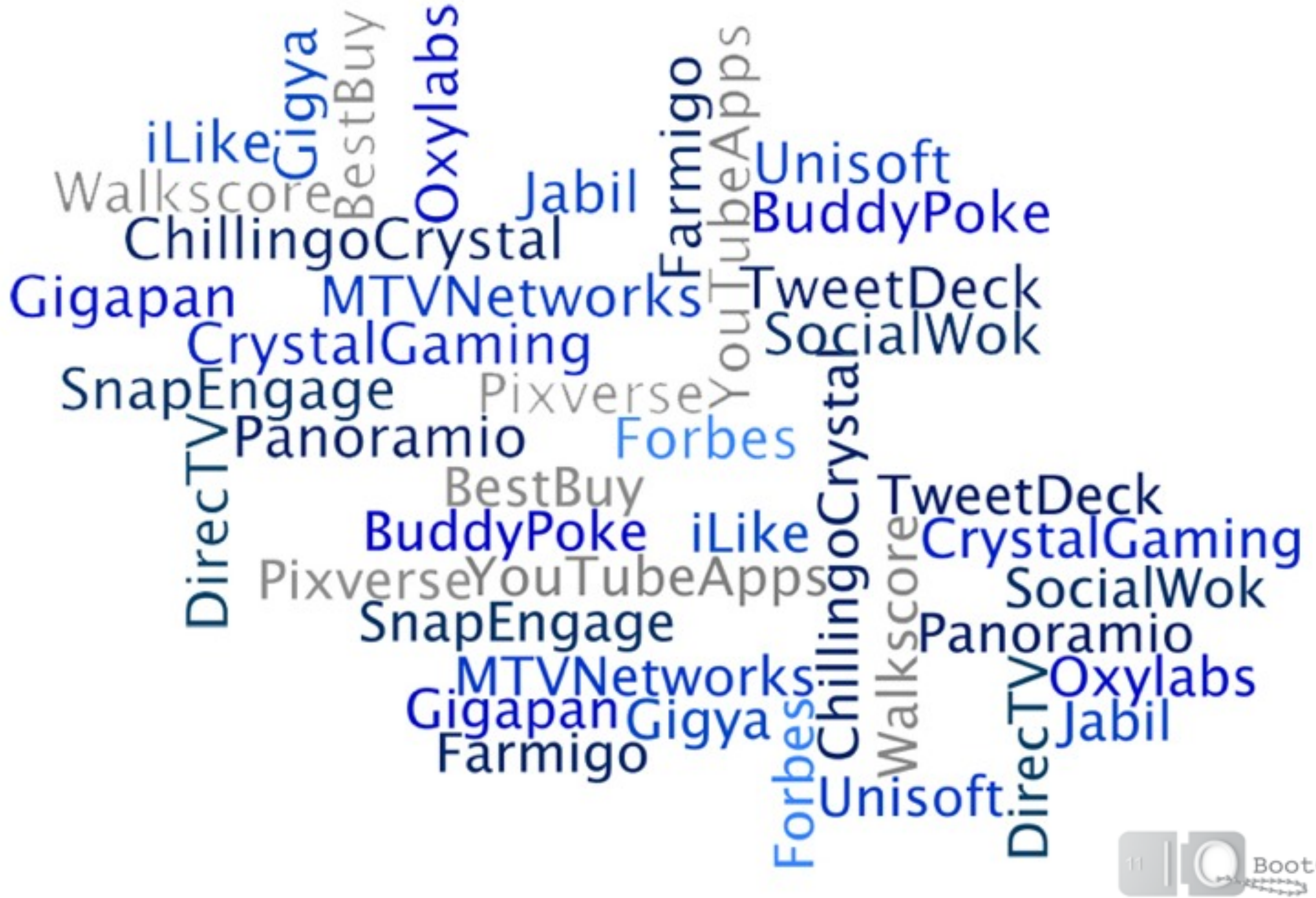
By the Numbers



1.5B

Pageviews per day

Some App Engine Partners



Factors to consider for picking a Cloud

- Price
- Type: Iaas, Paas, Saas
- Type of task: Apps, Big Data
- Public/Private/Hybrid
- Lock-In: Standards, Open Source

Issues to solve

- Cloud Interop: lack of standard
- Replication of Data across multiple Clouds
- Data privacy/integrity
  - encryption at rest
  - data auditing
- Trust, Culture of agility

Google Cloud Clients

- Chrome, HTML5
- ChromeBook, Device as a service $28/user/month
- Android: phone and tablets