How OpenX built a Scalable Global Digital Revenue Platform

Anthony Molinaro

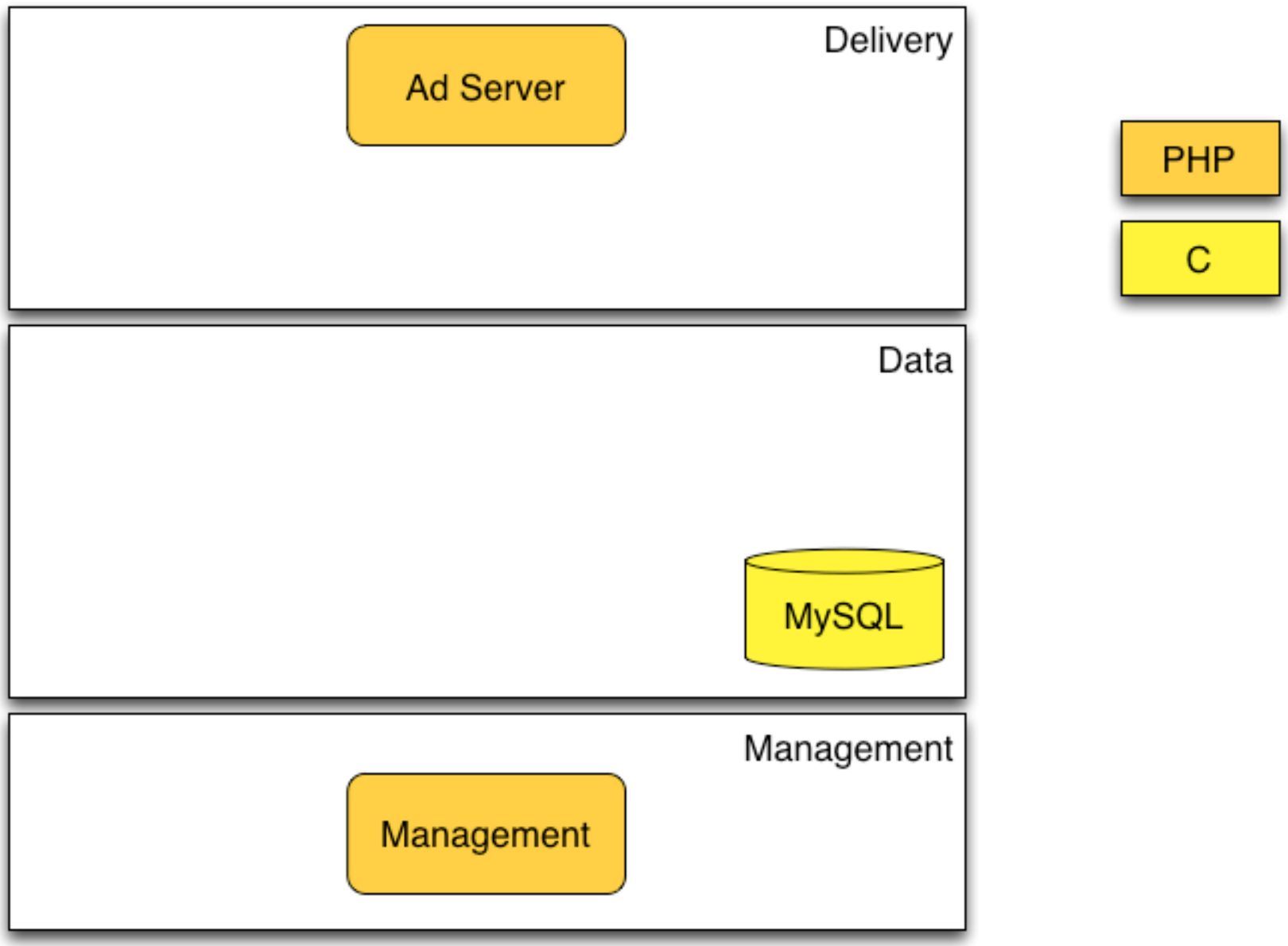
Talk Outline

- A brief history of the OpenX software stack
- How we came to use Erlang in that stack

1998-2007 Pre-History

- 1998 - phpAds
- 2000 - phpAds forked as phpAdsNew
- 2002 - phpAdsNew became OpenAds
- 2007 - OpenAds became OpenX
- Installed on 127,745 websites according to (http://trends.builtwith.com/topsites/Openads-OpenX)
- Built as a 2-tier app, PHP and MySQL

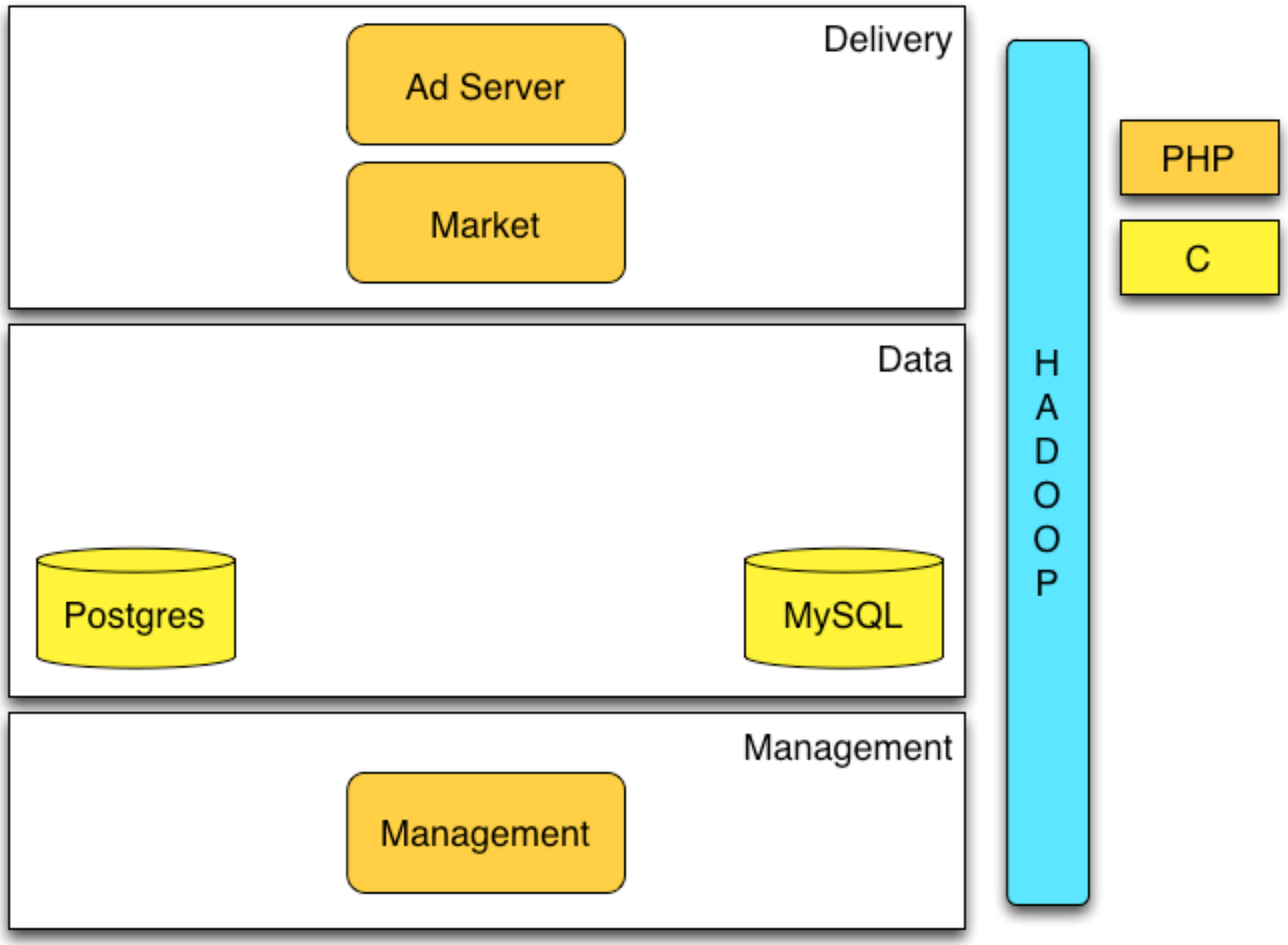
1998-2007 Architecture



2008 Incorporation

- OpenX is incorporated
- Headquarters moved to Pasadena
- Most development is still in Europe
- OpenX Market development begins
- LWES to Hadoop bridge created

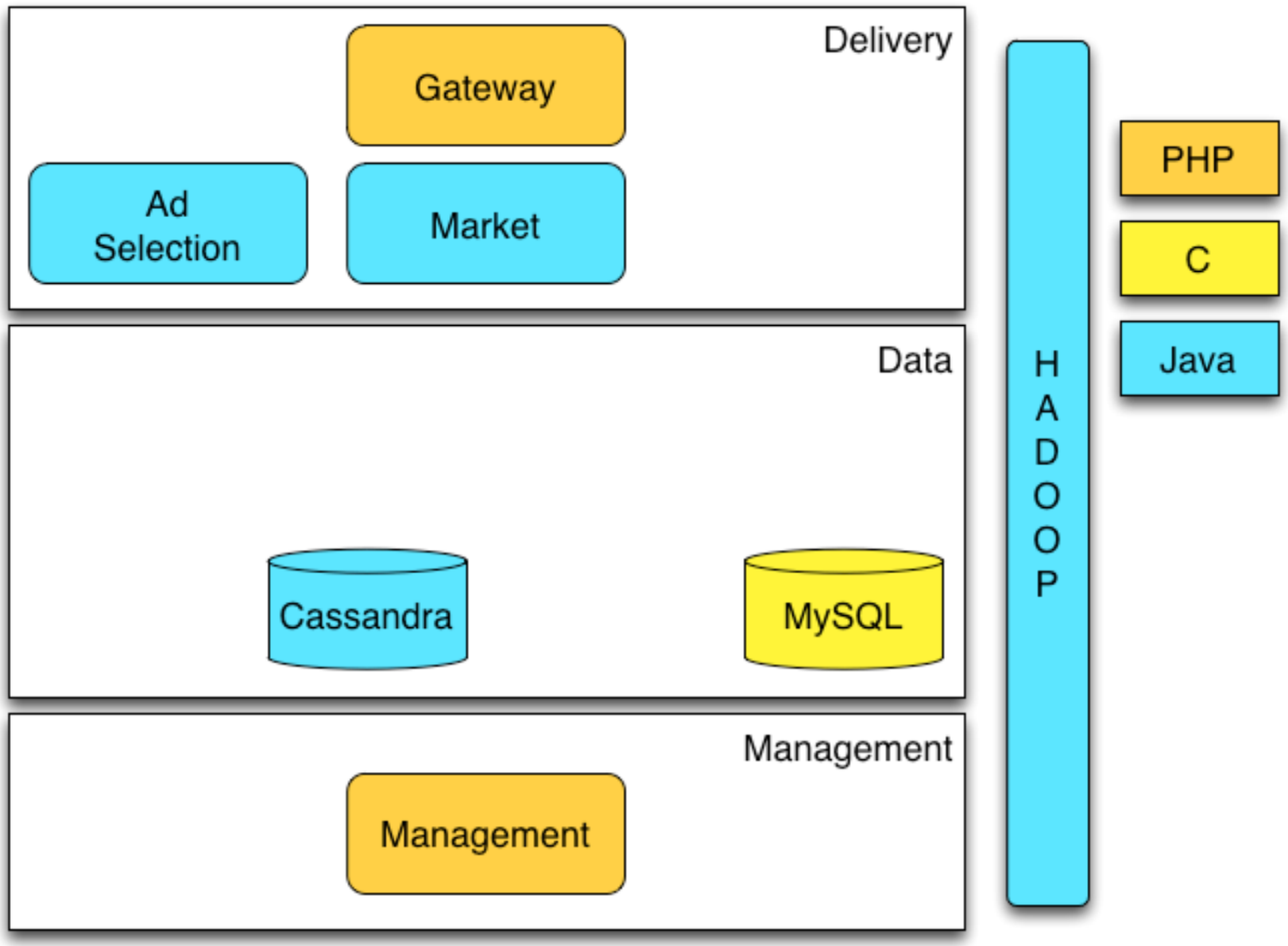
2008 Architecture



2009 Scaling the Market

- OpenX Market launches in April 2009
- PHP build on top of Postgres and MySQL
- Database writes on every request
- Requires many machines to scale
- Rewrite of Market and Ad Selection in Java
- Thrift between components and NoSQL for delivery data adopted

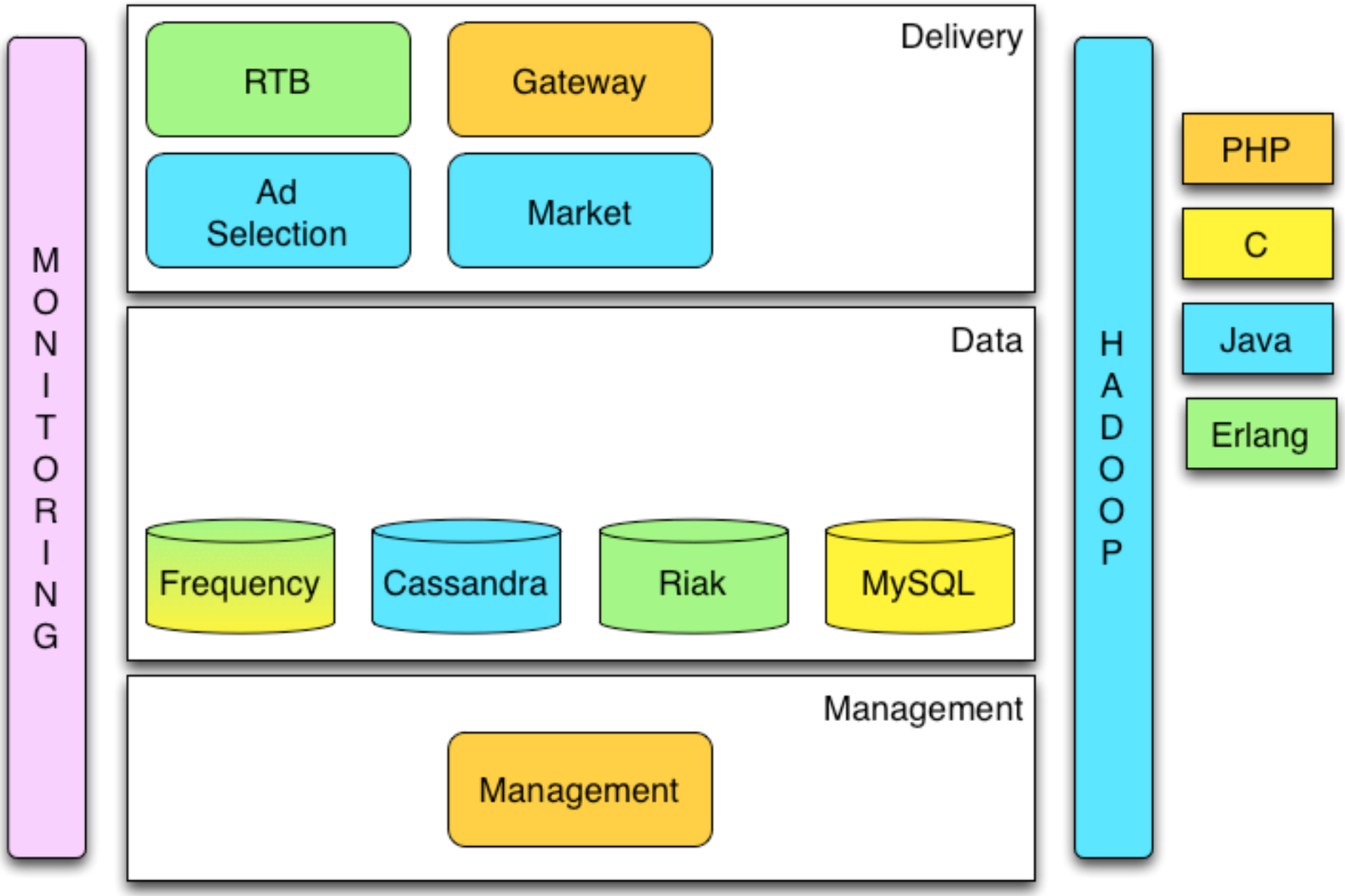
2009 Architecture



2010 Enter Erlang

- New features added to market
- RTB system
  - First system written in Erlang at OpenX
- Server side frequency
  - First Erlang/riak_core based system
- Riak added as a second NoSQL system
- Monitoring system deployed

2010 Architecture



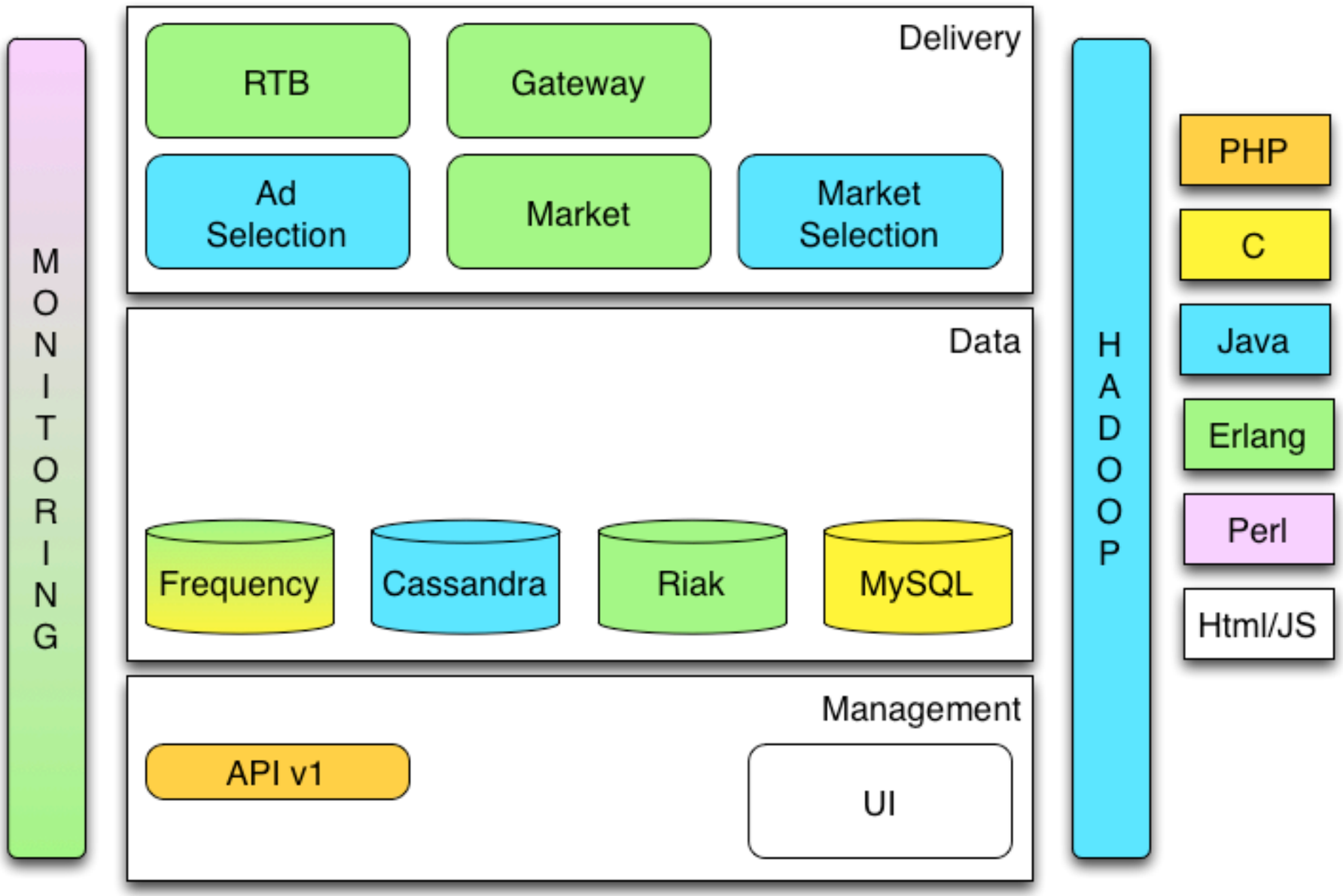
2011 OpenX Enterprise

- OpenX Enterprise 3.0 Rewrite
  - clean room rewrite
- Ad Delivery
  - 2 written in Erlang
  - 2 written in Java

2011 OpenX Enterprise

- Management
  - UI rewritten as HTML/Javascript
  - API rewritten in PHP
- Monitoring
  - Pieces rewritten in Erlang/CouchDB

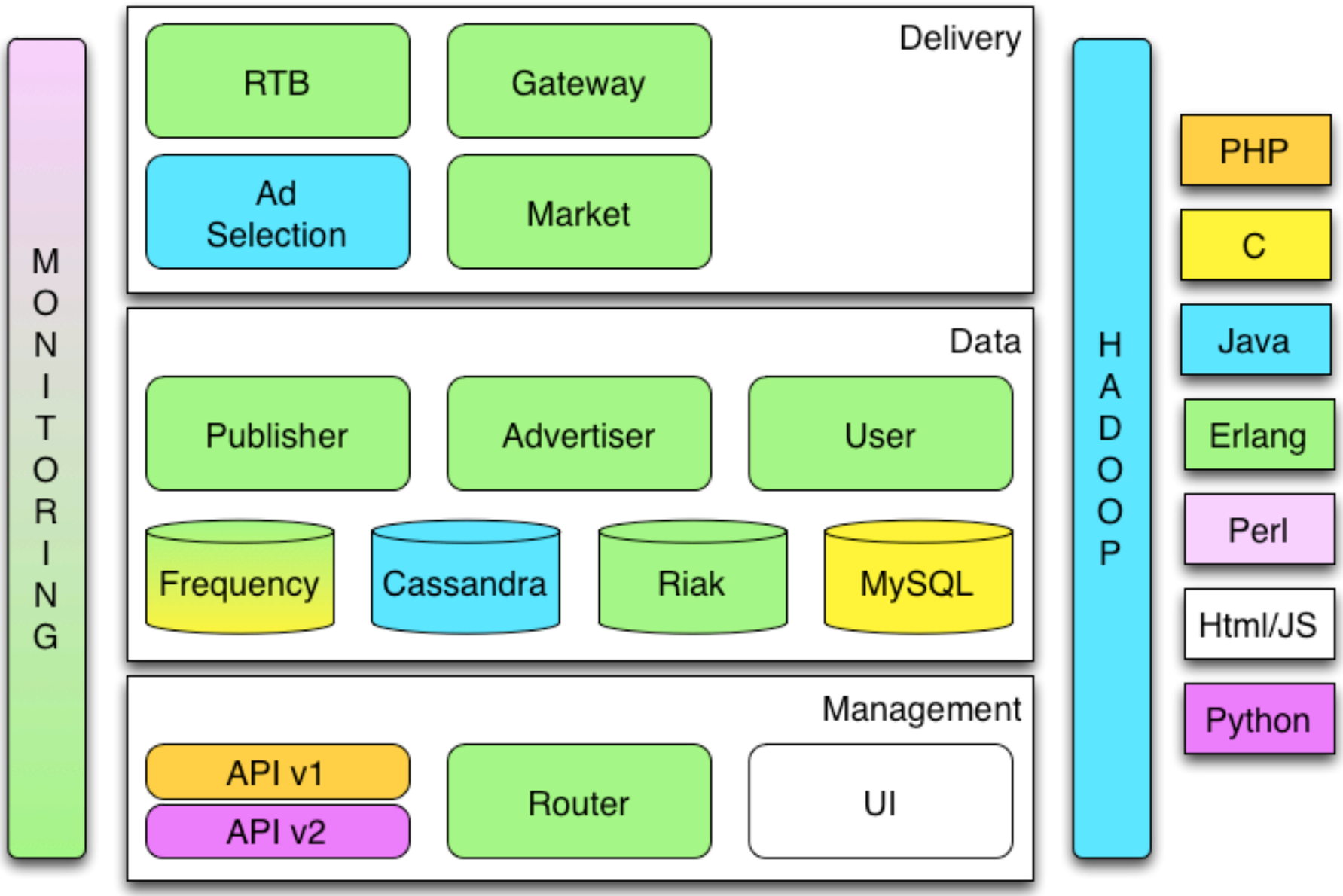
2011 Architecture



2012 Data Services

- Need to migrate off of Cassandra leads to 3 new services backed by Riak
  - User/Advertiser/Publisher Data Services all written in Erlang
- API Scalability issues lead to rewrite
  - 1 component in Erlang
  - 1 component in Python

2012 Architecture



Where did all these changes get us?

- 250+ Billion monthly ad transactions
- 12+ Billion daily bids
- Thousands of machines in 5 colos
- 265 employees
- $150M+ revenue in 2012

And for the languages

- 9 components in Erlang (probably closer to 11-12, but too many boxes)
- 2 components in Java (probably closer to 5-6 but again too many boxes)
- 1 in HTML/Javascript
- 1 in Python

How?

- Architecture
- Tools/Automation
- Evangelism

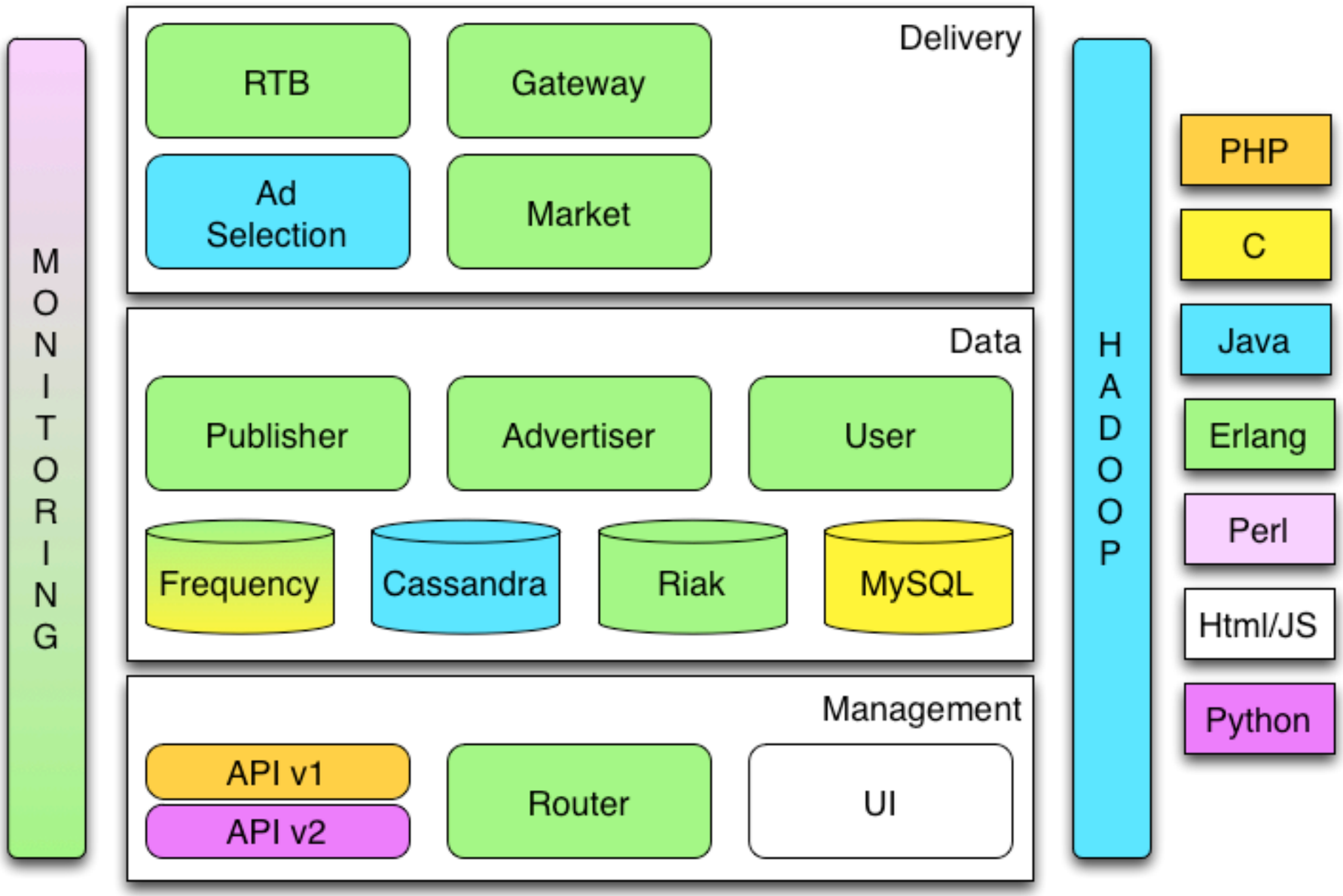
Architecture

- Cloud based
  - Generic hardware
  - Automated bootstrap and deployment
  - Package oriented development
  - Fault tolerant

Architecture

- Service based
  - Loosely coupled
  - Single purpose components
  - Pools of components
  - Polyglot

2012 Architecture



Deployment

- Started with a system written at Google called slack
  - Use rsync to sync files to a machine
    - Overlay slack files onto file system
    - Provides roles and subroles
- Used for bootstrap, deployment and configuration

Packaging

- Package Oriented Development
  - Package for your OS (rpm/deb/etc)
    - Launches and rollbacks are usually a single command line
    - Developers build and exchange packages
    - Semantic Versioning

Packaging for Developers

- framewerk
  - Pluggable build system, but currently most templates based on autotools
  - Provides templates for code layout
  - Targets for compiling, testing and packaging
  - Common commands across languages
  - Enforces versioning and reproducibility

Fault Tolerance Through Load Balancing

- Client side haproxy
  - Service's configured to connect to a single port on localhost
  - Haproxy manage's connections
  - Haproxy fail over allows for easy rolling upgrades of any service
  - Challenge is machine list updates

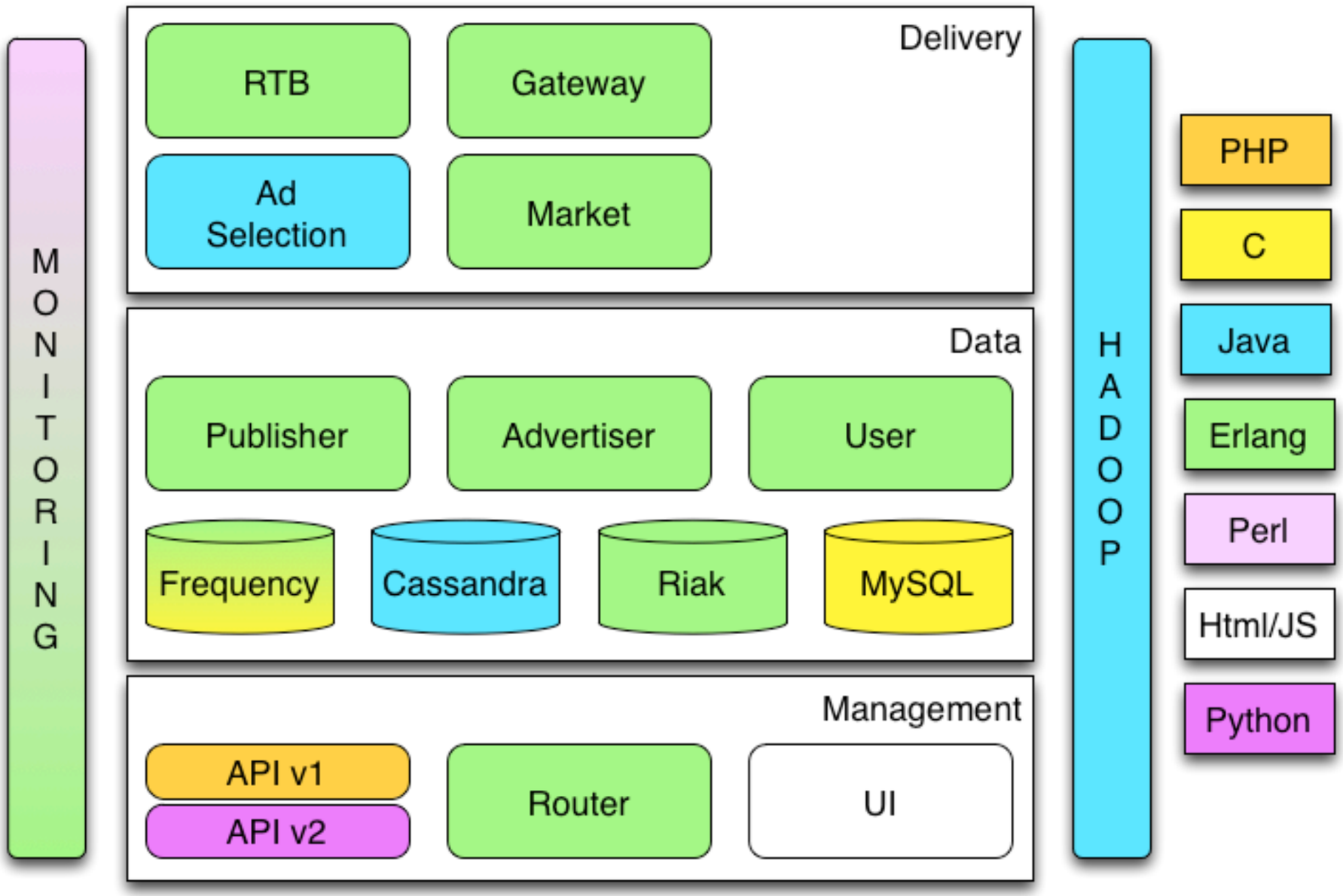
Load Balancing Part 2

- Persistent connections
  - gen_server_pool
    - Turn any gen_server into a pool with minimal effort
    - Used for thrift and riak connections
- Similar java pool created, but less generic

Cross Language Communication

- thrift
  - RPC between most components
- protobuf
  - RTB and Riak
- lwes
  - Logging and monitoring

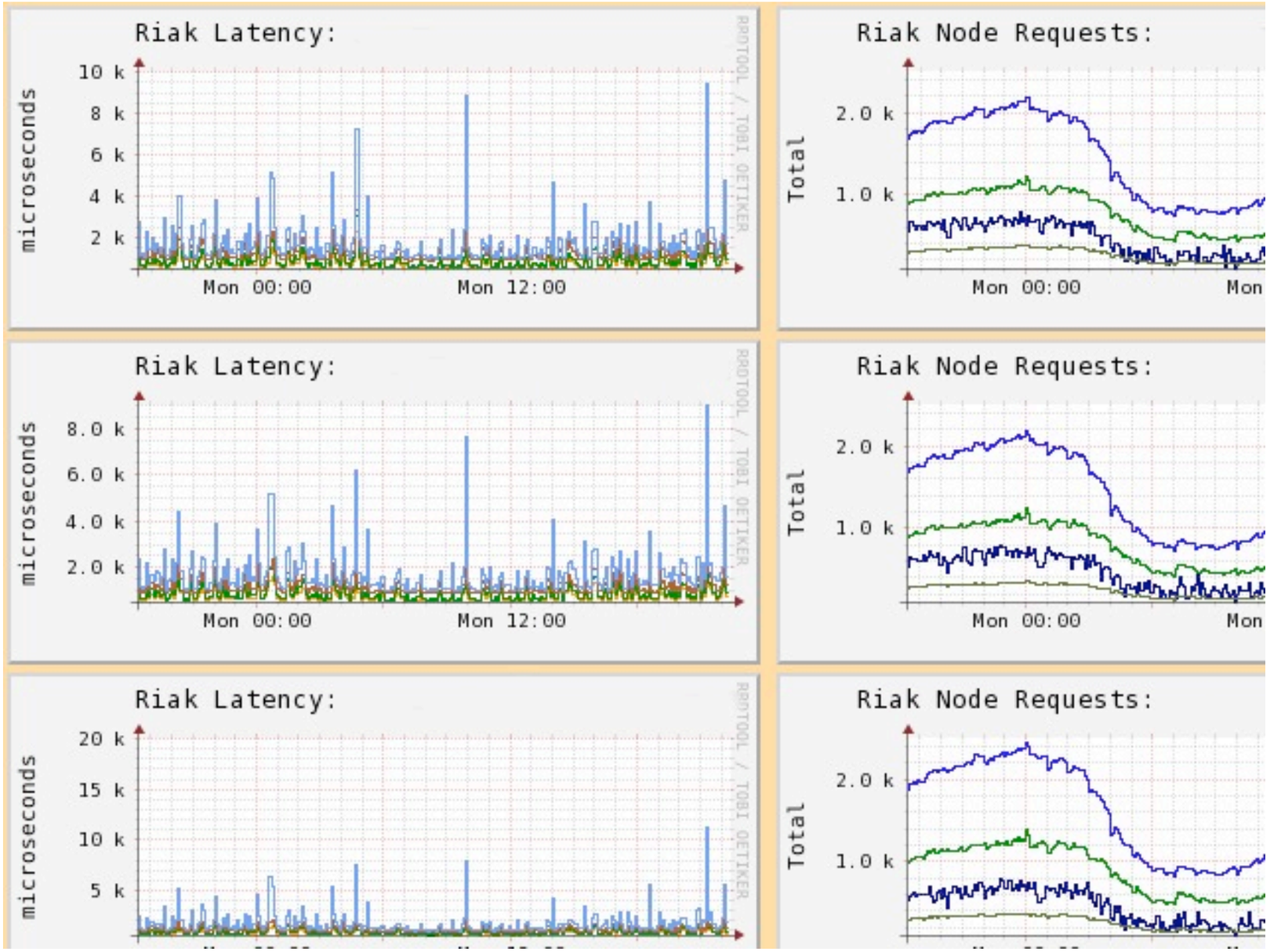
2012 Architecture



Monitoring

- munin
- mondemand
  - Cross language via lwes
  - Stats are easy, just add a line to your application then get an rrd you can graph
  - Traces of requests allow for troubleshooting and QA

Mondemand Stats



Mondemand Traces

Trace for owner1:test1

⊞Mar 08, 2013 15:23:16.0 - gateway (host1) - Trace Starting
⊞Mar 08, 2013 15:23:24.0 - user_server (host2) - getPublisherSegments thrift called with
⊟Mar 08, 2013 15:23:24.0 - user_server (host2) - reading publisher segment data from ri
  ⊟segments
    12345 : 1364405340
    67890 : 1364421360
    88888 : 1365292252
⊞Mar 08, 2013 15:23:24.0 - gateway (host1) - getPublisherSegments thrift call responde
⊟Mar 08, 2013 15:23:24.0 - gateway (host1) - calling requestAds of ad selection
  ⊟request
    instance_id : feedabbadeadbeef
    ⊞request_context
    ⊟selection_ids
      ⊟0
        ⊞context
        id : 220172
        type : AdUnit
      user_id : adecafcafeadecade
    request_url : http://foo.openxenterprise.cor
⊞Mar 08, 2013 15:23:25.0 - ad selection (host3) - AdResponses
⊟Mar 08, 2013 15:23:25.0 - gateway (host3) - ad selection requestAds responded
  ⊞response
    response_millis : 17
⊟Mar 08, 2013 15:23:25.0 - gateway (host4) - call ad selection for recordEvent of impres
  ⊞request

Evangelism

- If possible ‘fix’ the game via architecture and tooling choices
- Find a project that showcases the technology
- Make sure the project succeeds
- Make sure to share work
- Make it easy for others

Making Erlang Easy

- Developing and Package (framewerk)
  - fw-template-erlang
    - erlrc integration
  - fw-template-erlang-rebar
    - turn thirdparty erlang projects into packages a single command line
    - best for NIFs and ports

Making Erlang Easy

- Running
  - erlrc - integrates with packaging system to create boot scripts and hot code load on package installation/removal
  - erlstart - start/stop and connect to an erlang node
  - erlnode - integrate with Red Hat services

Links

- https://code.google.com/p/slack/
- https://github.com/dukesoferl/fw/
- https://github.com/openx/gen_server_pool
- http://www.lwes.org
- http://www.mondemand.org
- https://github.com/dukesoferl/erl(rc|start|node)

Questions?

- anthony.molinaro@openx.com
- https://github.com/djnym