Introducing Erlang to OpenX

Anthony Molinaro

What is OpenX?

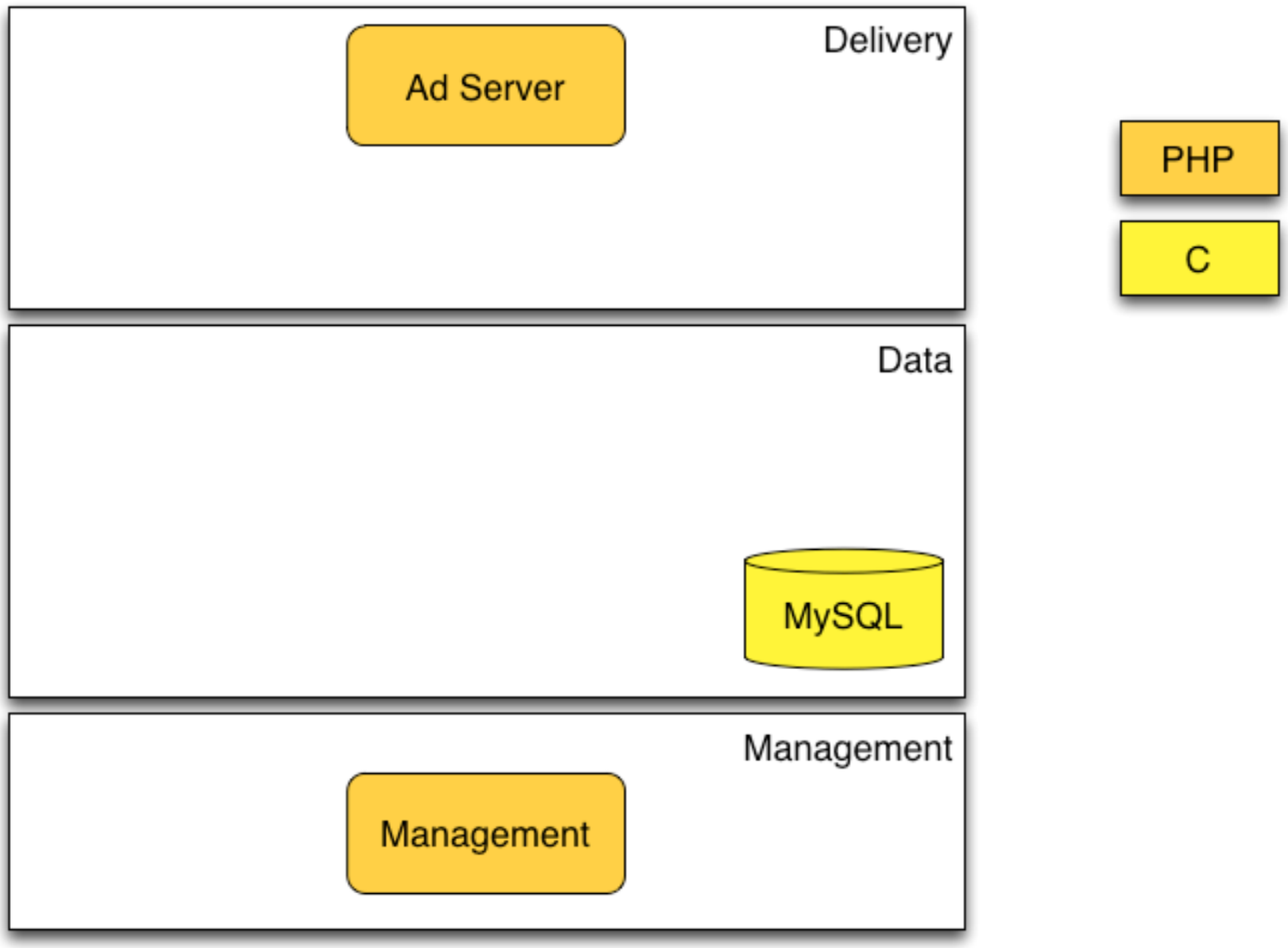
What is OpenX?

- An Open Source PHP ad server.

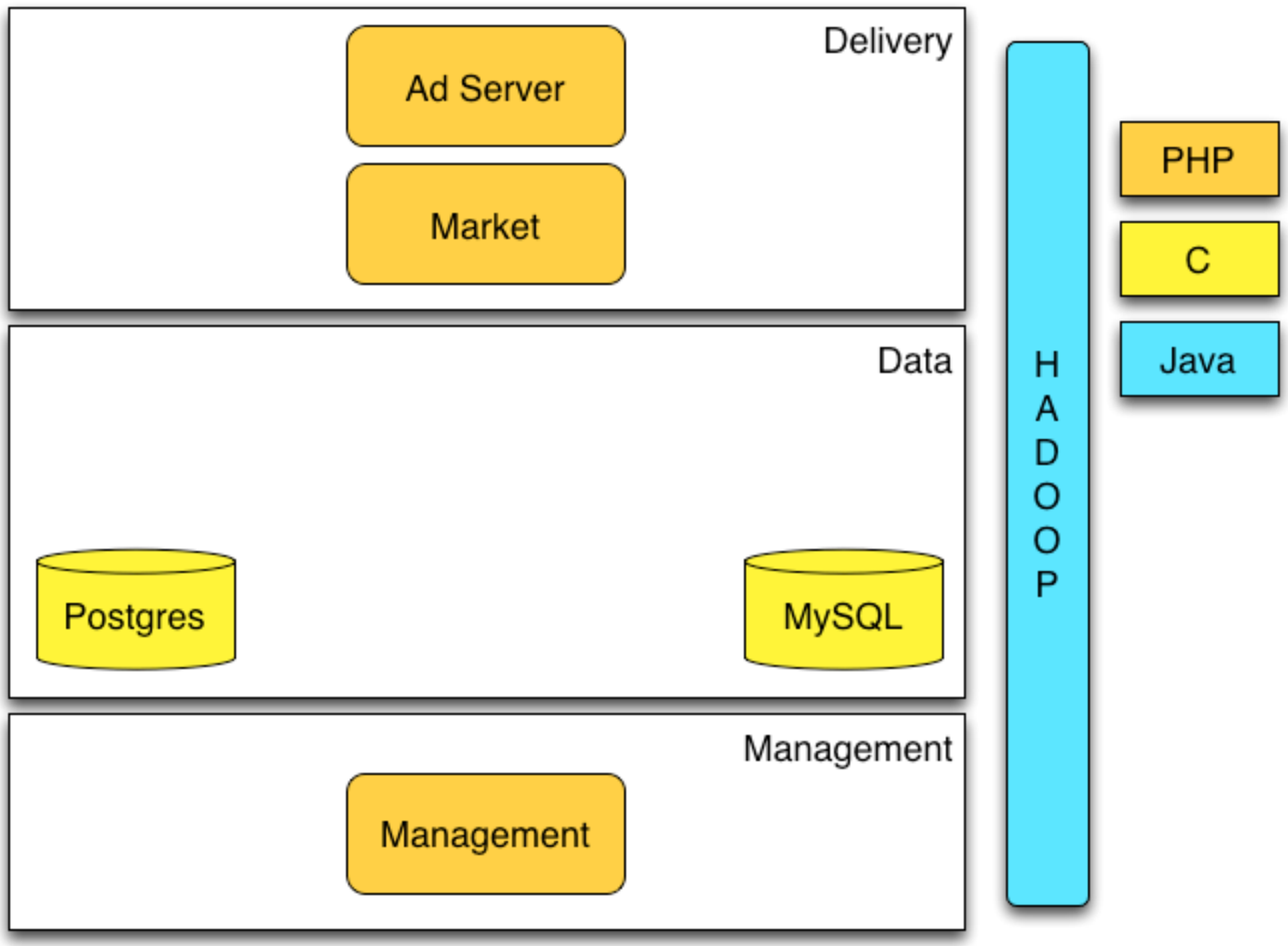
What is OpenX?

- An Open Source PHP ad server.
- A global company with a SAAS enterprise ad server featuring an integrated ad exchange written largely in Erlang.

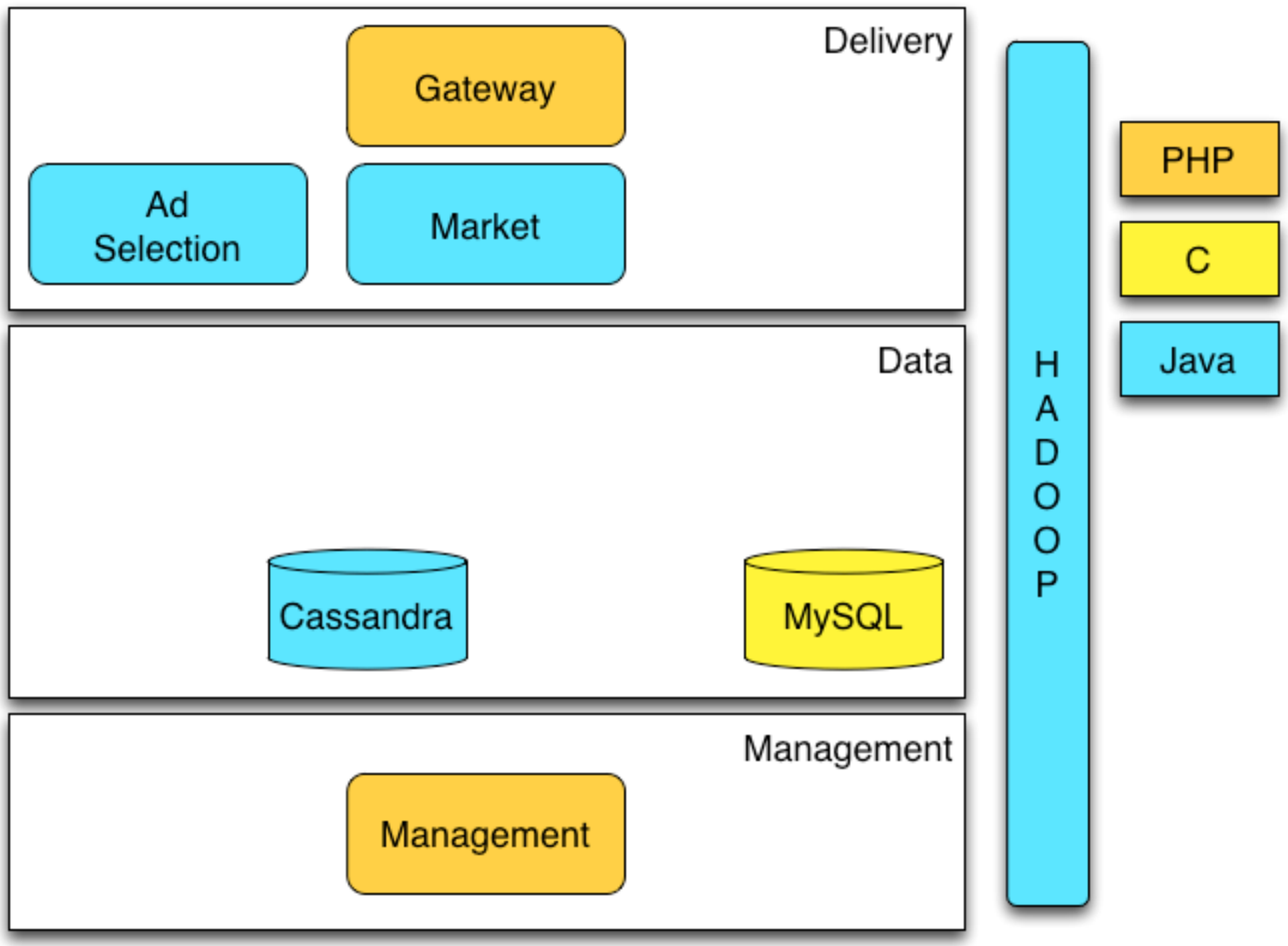
1998-2007



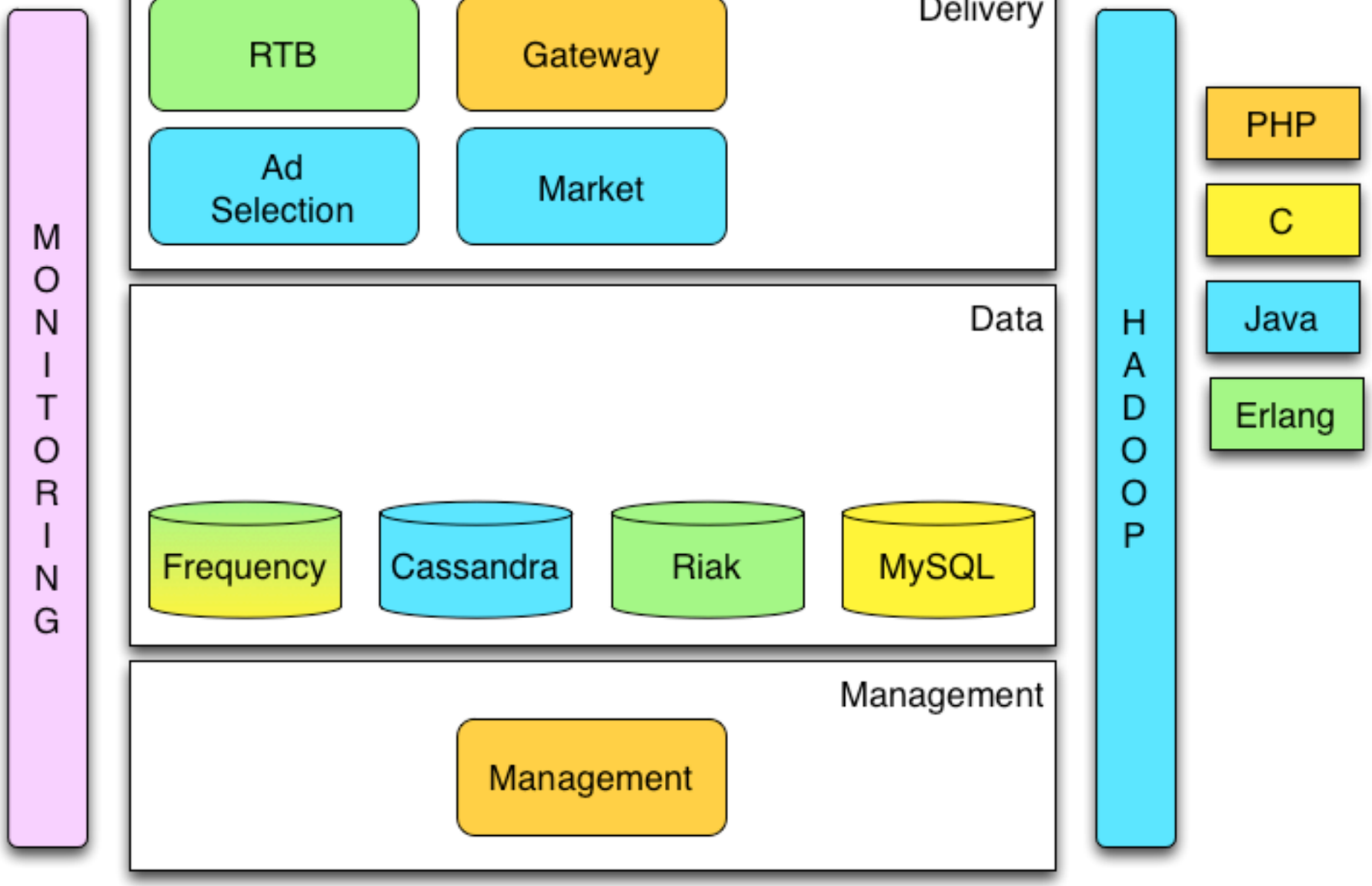
2008


Delivery
Ad Server
Market
Data
Postgres
MySQL
Management
Management
HADOOP
PHP
C
Java

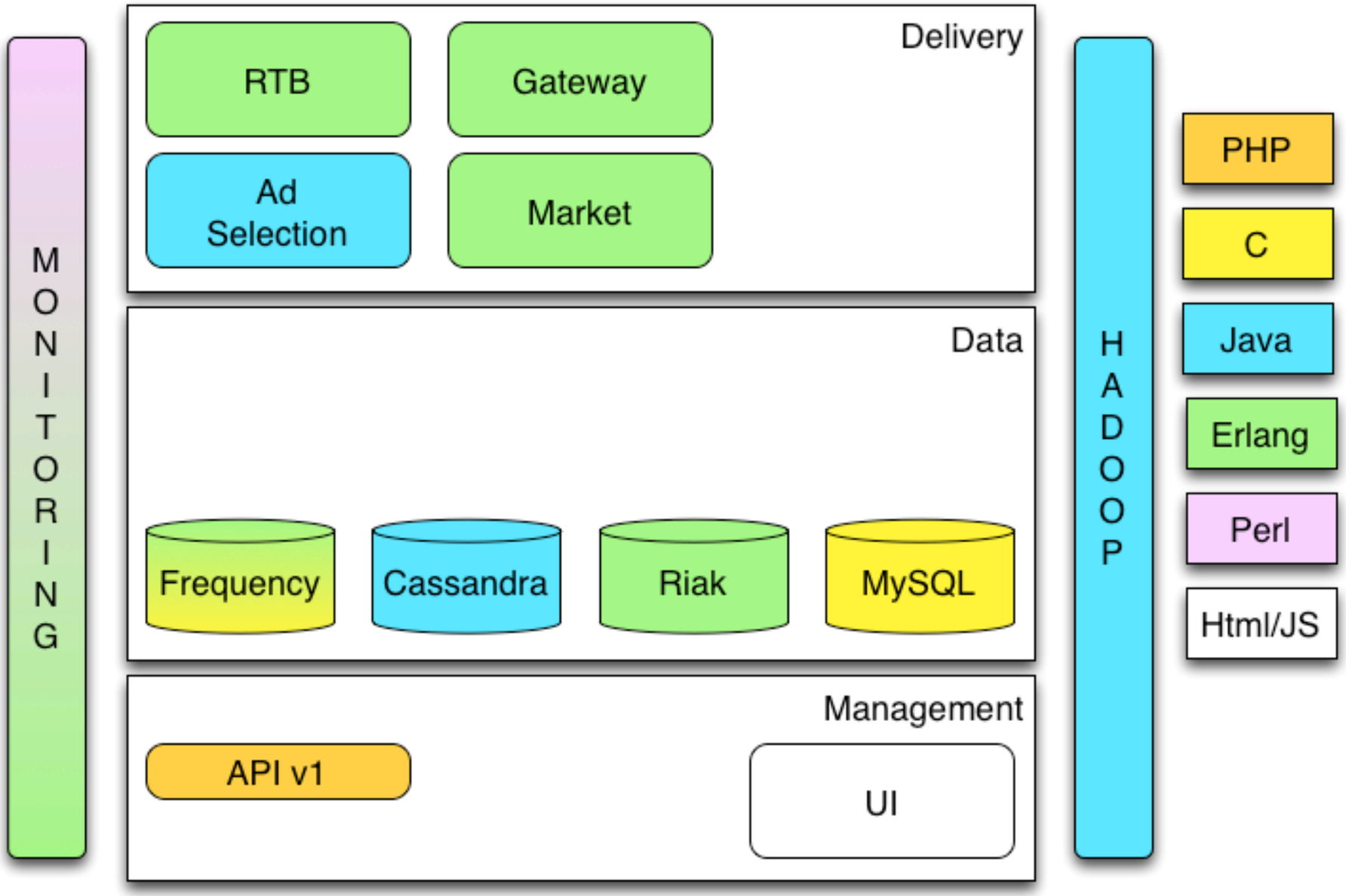
2009


Delivery
Gateway
Ad Selection
Market
Data
Cassandra
MySQL
Management
Management
HADOOP
PHP
C
Java

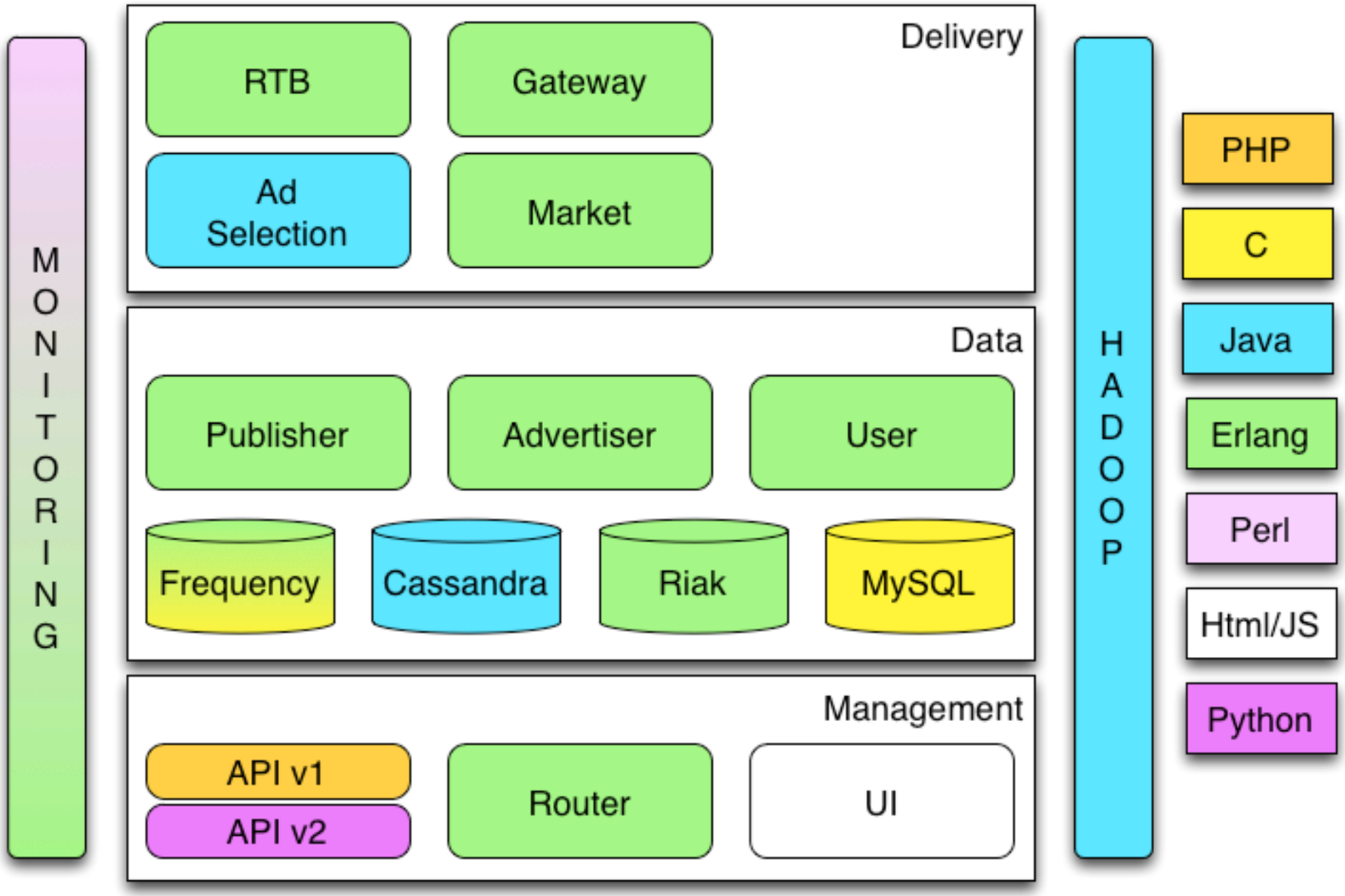
2010


MONITORING
Delivery
RTB
Gateway
Ad Selection
Market
Data
Frequency
Cassandra
Riak
MySQL
Management
Management
HADOOP
PHP
C
Java
Erlang

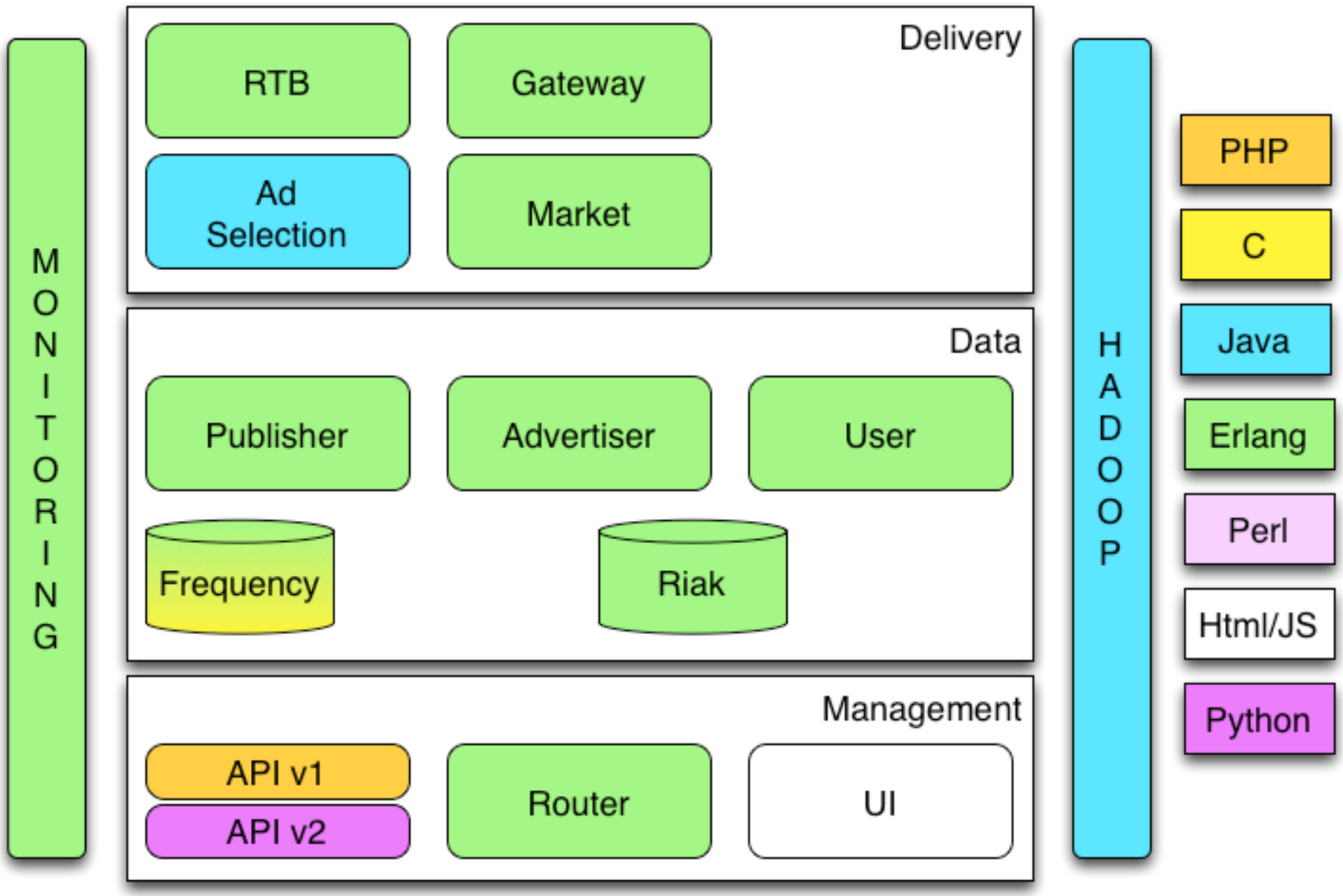
2011


MONITORING
Delivery
RTB
Gateway
Ad Selection
Market
Data
Frequency
Cassandra
Riak
MySQL
Management
API v1
UI
HADOOP
PHP
C
Java
Erlang
Perl
Html/JS

2012


MONITORING
Delivery
RTB
Gateway
Ad Selection
Market
Data
Publisher
Advertiser
User
Frequency
Cassandra
Riak
MySQL
Management
API v1
API v2
Router
UI
HADOOP
PHP
C
Java
Erlang
Perl
Html/JS
Python

Early 2013


MONITORING
Delivery
RTB
Gateway
Ad Selection
Market
Data
Publisher
Advertiser
User
Frequency
Riak
Management
API v1
API v2
Router
UI
HADOOP
PHP
C
Java
Erlang
Perl
Html/JS
Python

What languages are currently used?

- 14-15 services in Erlang
- 7-8 services in Java
- 2 services in HTML/Javascript
- 2 services in Python
- 2 services in PHP

How did a PHP shop end up with so much Erlang?

How did a PHP shop end up with so much Erlang?

- Architecture

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

How did a PHP shop end up with so much Erlang?

- Architecture
- Tools

Cross Language Communication

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

Cross Language Build/ Packaging

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

How did a PHP shop end up with so much Erlang?

- Architecture
- Tools
- Evangelism

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 - tools!

Making Erlang Easy

- Developing and Packaging (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

Challenges

- Hiring is almost impossible.
- Operations have unfamiliar systems to manage
- Developers have new concepts and patterns.
- You are often breaking new ground.

How did Erlang help OpenX developers?

- Fits the service model really well.
- "Let it crash" philosophy and supervision trees meant crashes were rarely fatal.
- The VM scales really well across multiple cores.
- Functional aspects lead to easy to test and easier to read code and often faster to develop

How did Erlang help OpenX?

- 250+ Billion monthly ad transactions
- 12+ Billion daily bids
- Thousands of machines in 5 colos
- ~300 employees
- $150M+ revenue in 2012

Questions?

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