StoryCloud

Server Architecture (version 1)
Barry Shapira

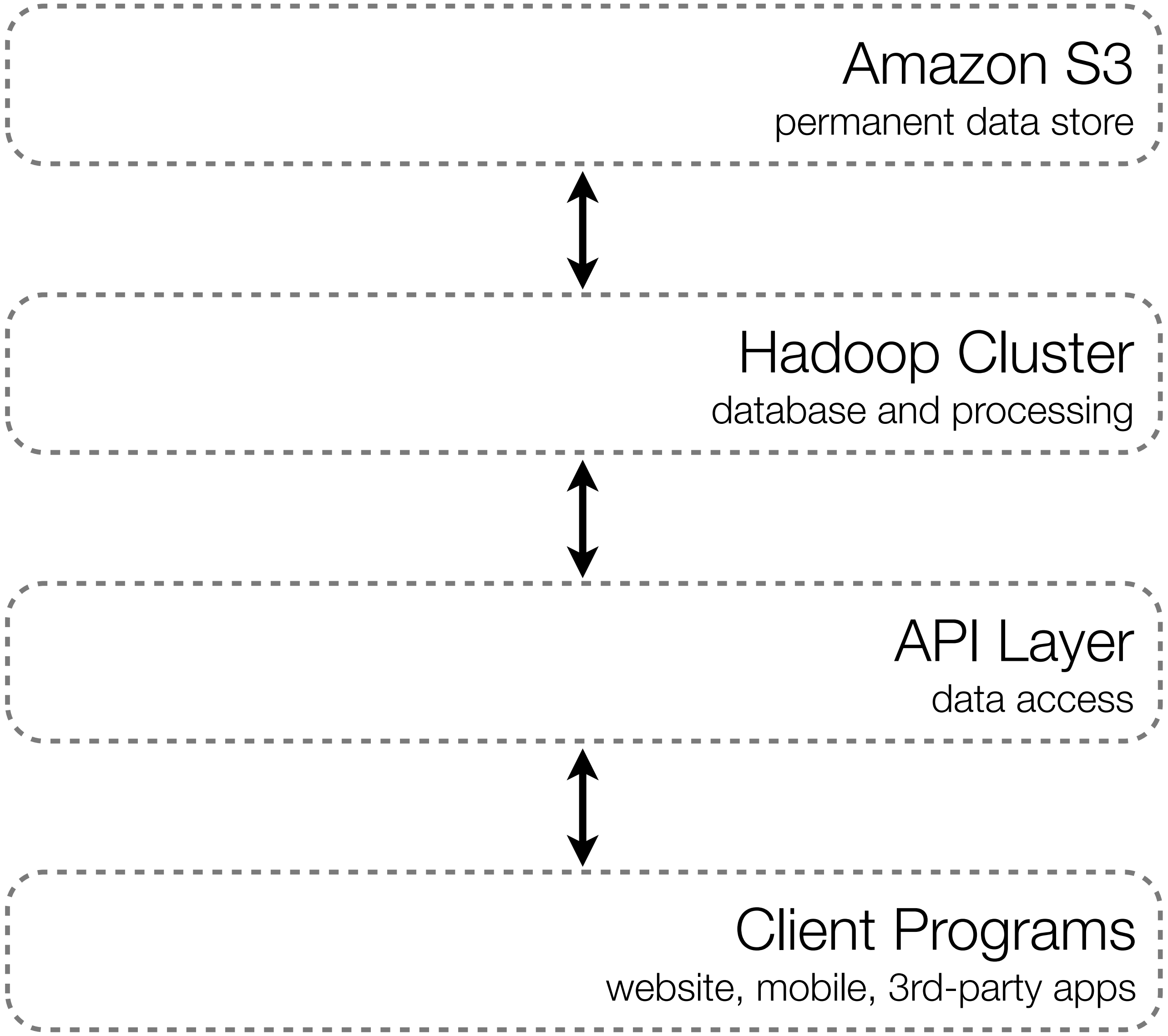
Monday, December 16, 2013

StoryCloud Architecture

- Architecture overview
- Backup strategy
- Development plan

Architecture Overview

independently scalable layers



Amazon S3

permanent data store

Hadoop Cluster

database and processing

API Layer

data access

Client Programs

website, mobile, 3rd-party apps

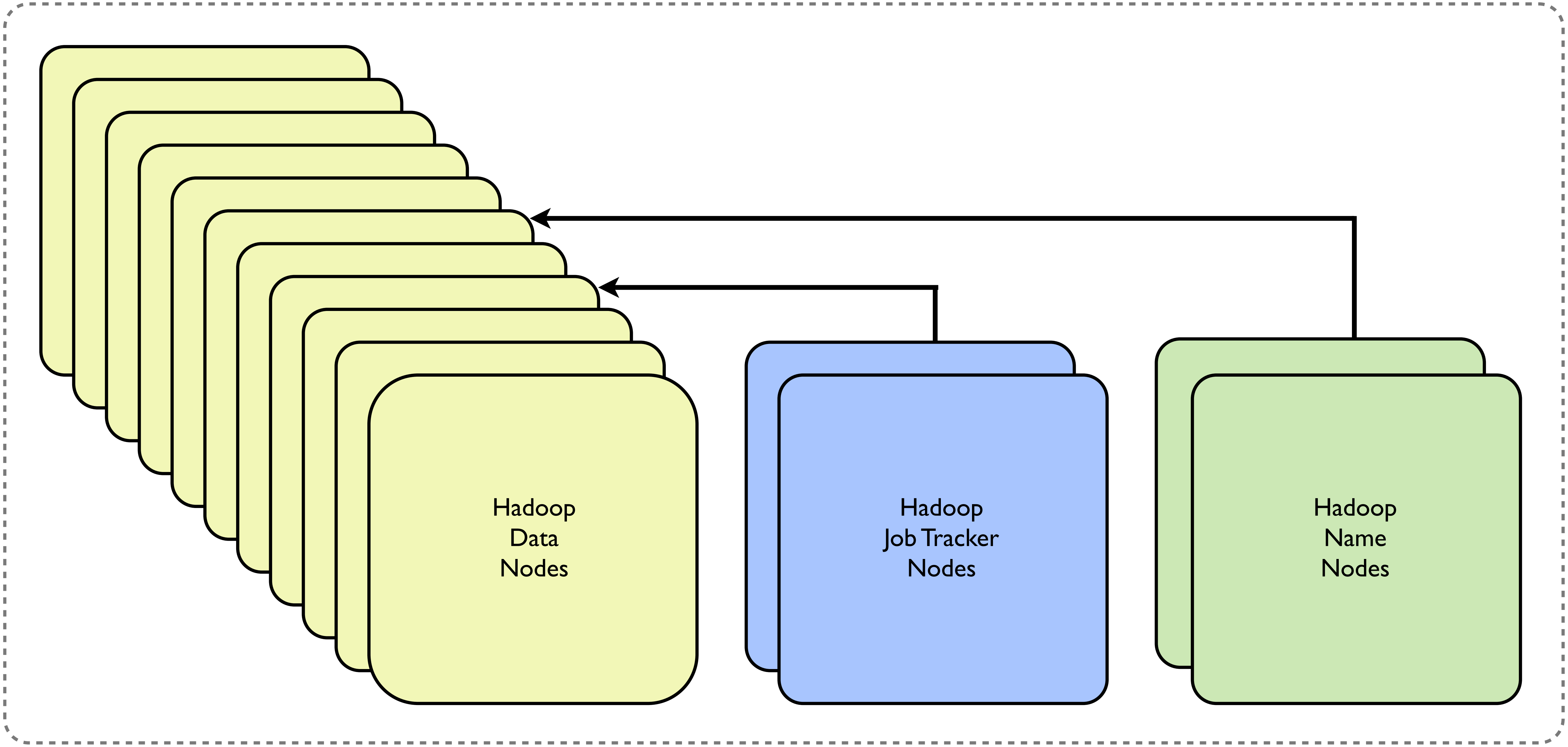
Amazon S3

permanent data store

- Infinitely scalable
- About $0.80 per gigabyte per month
- 99.9999999% durability and 99.99% availability
- Used for client file storage and database backups

Hadoop Cluster

database and processing



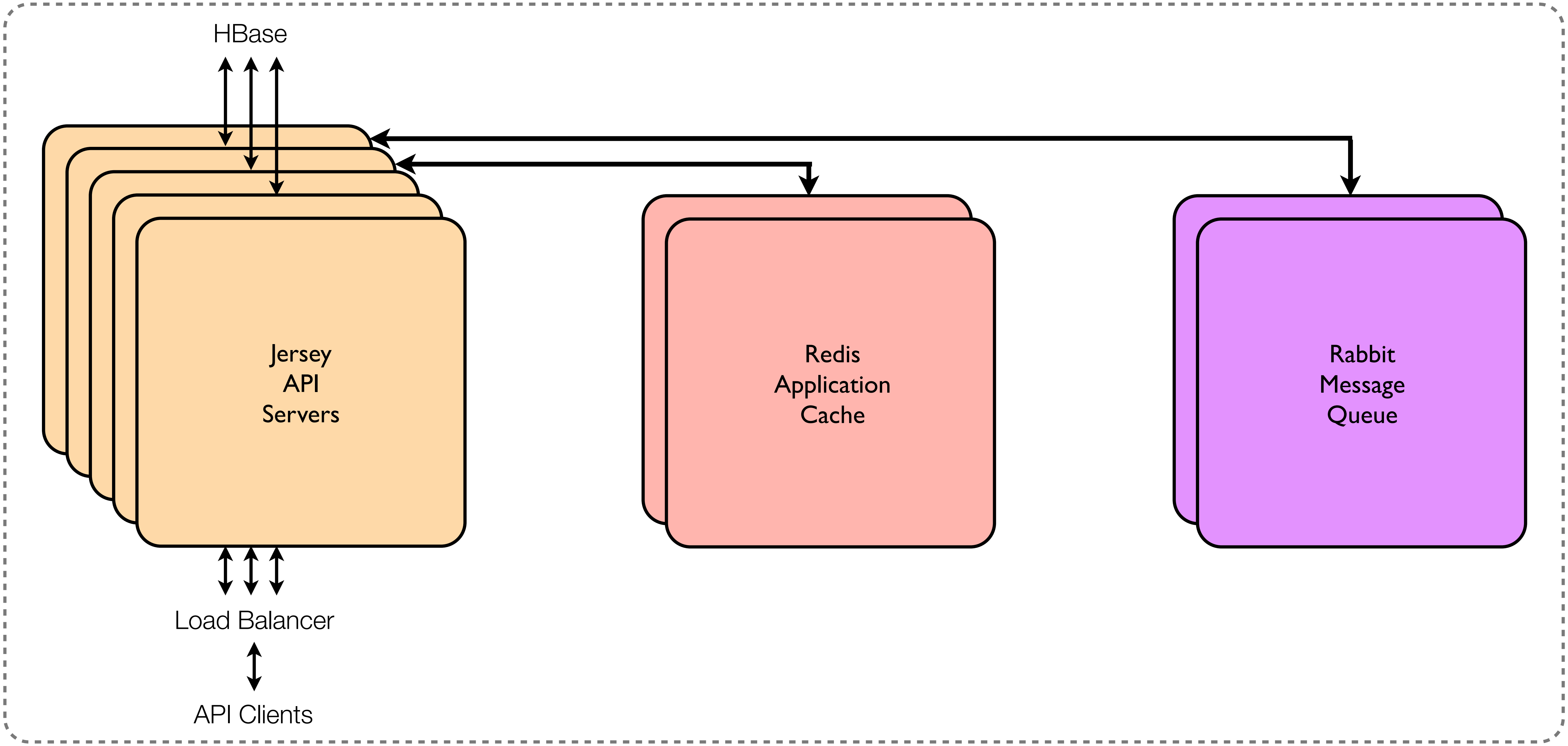
Hadoop Cluster

database and processing

- The industry standard for “big data” storage and processing
- Scales to thousands of nodes for petabytes of storage and teraflops of processing power
- HBase (the hadoop database) has a flexible schema and fast random access for “billions of rows and millions of columns”
- Large “ecosystem” of related software for business intelligence, machine learning, batch processing, etc.
- Can be complicated to set up but the Cloudera Distribution with Hadoop does most of the heavy lifting

API Layer

data access



Jersey

API servers

- Written in Java for native access to the Hadoop cluster
- The reference implementation of JAX-RS, the Java REST standard
- Huge support community and tons of plugins
- Feature-rich and reasonably fast
- Stateless, share-nothing architecture ensures horizontal scalability

Redis

application cache

- Application cache stores frequently requested items, lowering API request times and database load
- Easy key-value access model
- Can store complex data types
- Fast and stable

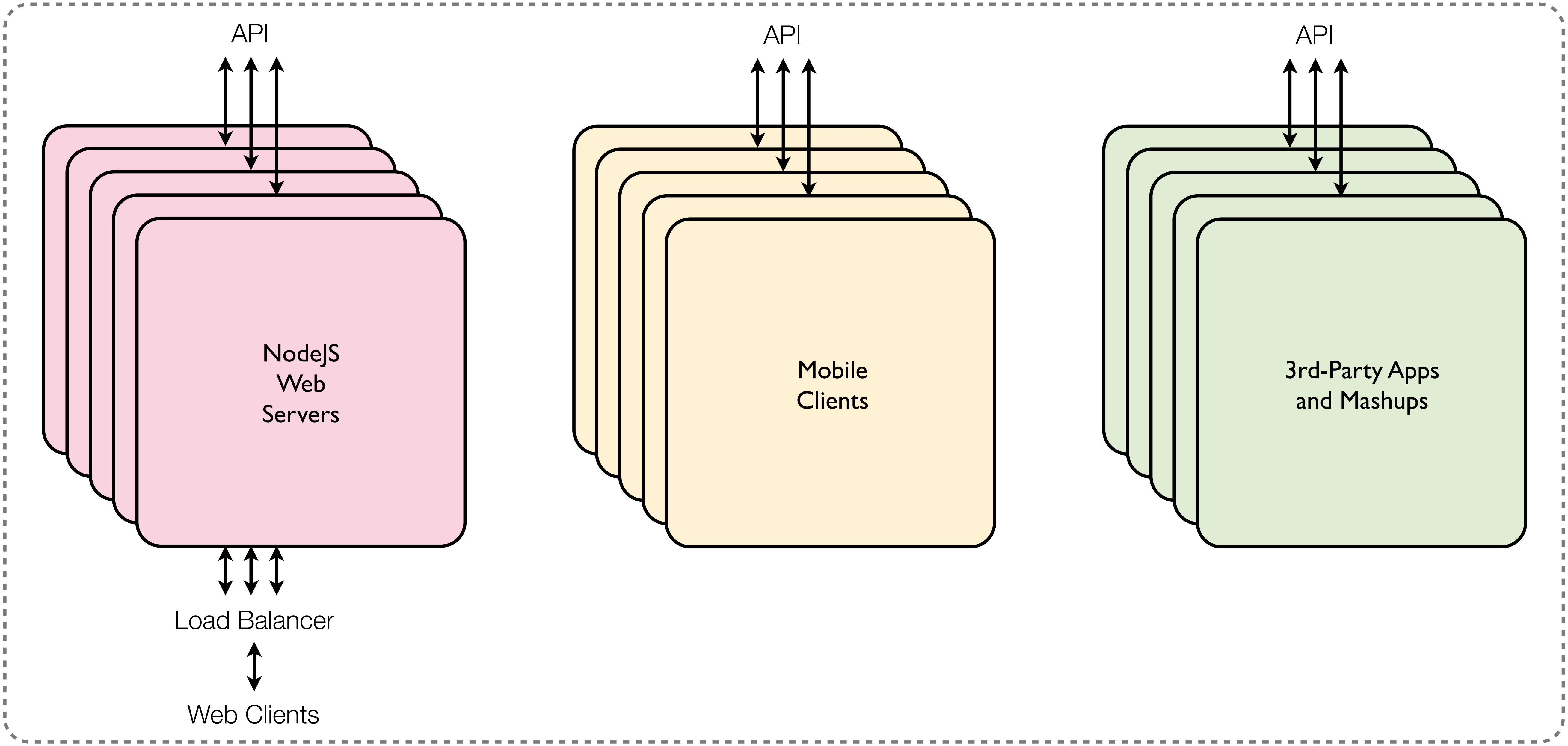
Rabbit

message queue

- Message queue handles long-running tasks (like image processing or communicating with external APIs)
- Keeps response time and API server load low
- Can be clustered or federated for scalability and reliability
- Popular, robust, and easy to use

Client Programs

website, mobile, 3rd-party apps

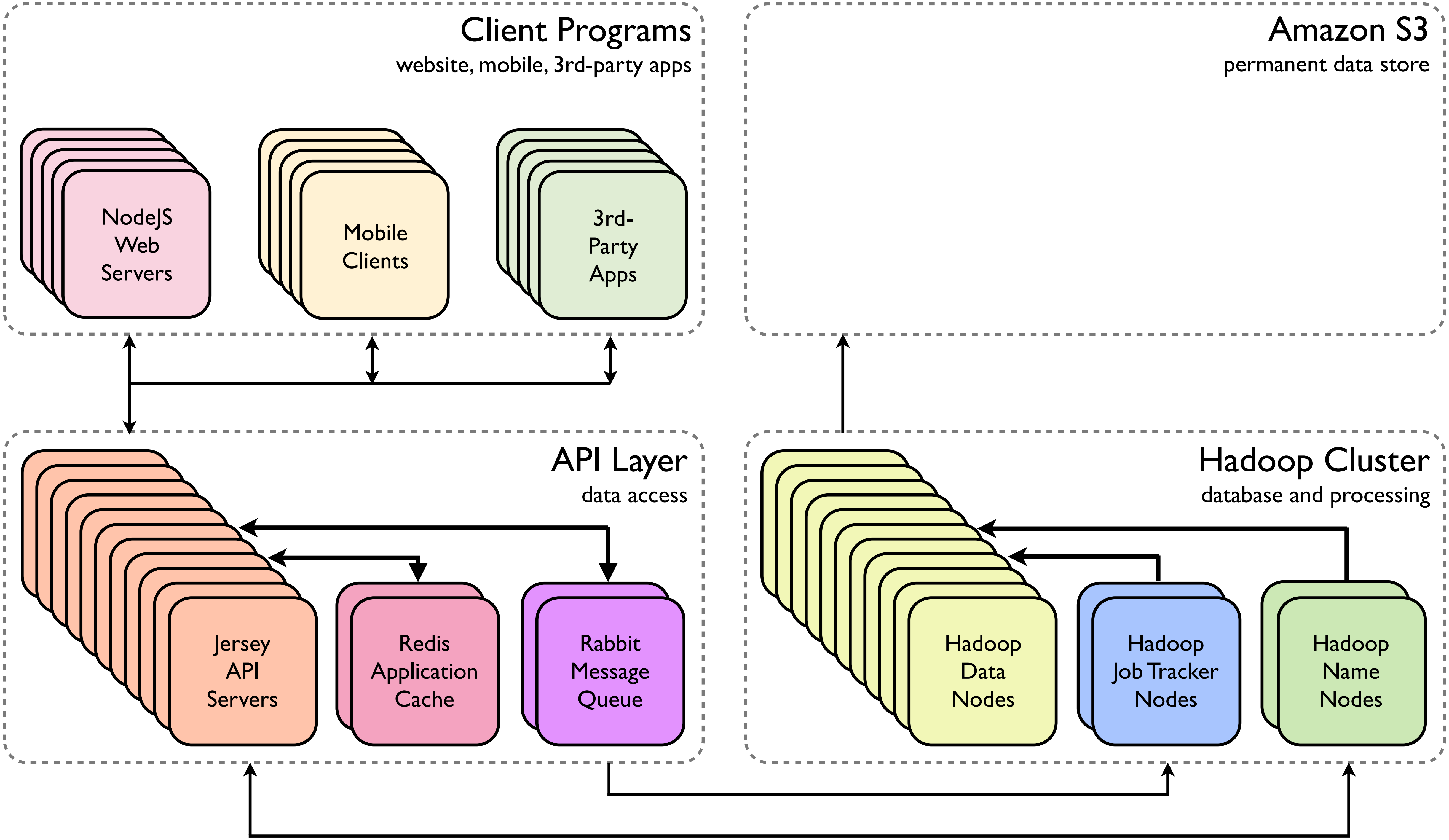


NodeJS

web server

- Asynchronous event loop allows huge numbers of client connections
- API responds in JSON, which Node reads natively
- Client-side browser code is necessarily written in JavaScript. Having the middle-tier server in the same language makes life easier for front-end programmers.
- Stateless, share-nothing architecture ensures horizontal scalability

How It All Fits Together



Backup Strategy

Amazon S3 and Github

- User files
  - Stored directly in S3.
- Hadoop
  - Primary and backup name nodes save to RAID10 EBS volumes which are backed up hourly to S3.
  - HBase is replicated using HDFS and backed up regularly to S3.
- Codebase
  - Developed using git version control and stored on Github.

Development Plan

- Database
- API
- Website and Mobile

Development Plan

database

- Develop initial database schema (late December)
- Set up Amazon EC2 servers (mid January)
- Install Cloudera Distribution with Hadoop (late January)
- Test cluster stability (early February)
- Requirements:
  - 10 Amazon EC2 servers (part-time usage)
  - Hire a Hadoop expert

Development Plan

API

- Design API for user management, document storage, timeline display, and social media sharing (late December)
- Set up Jersey servers with “Hello World” application (mid Feburary)
- Connect API servers to database (late February)
- Write API version 1 (early March)
- Requirements:
  - 4 Amazon EC2 servers (part-time usage)
  - Hire a Java expert

Development Plan

website

- Set up Node servers (late February)
- Design initial website version (March)
- Set up account creation and login (early March)
- Allow saving files and displaying them in a timeline (mid March)
- Allow publishing/sharing to social media (mid April)
- Requirements:
  - 2 Amazon EC2 servers (full-time usage)
  - Hire a UI/UX designer and 2 HTML/CSS/JavaScript experts