SCIENCE

&

MUSIC

PLANNING

- **Tuesday 29/11**
  - Talk & QA: *Music & Web - Architecture & Technology Overview*
  - Lab Work 1: Working with APIs
  - Lab Work 2 & 3: Introduction to Pandas & data analysis
- **Tuesday 6/12**
  - Talk & QA: *Music & Big Data - Overview of challenges & technologies*
  - Lab Work 4: Introduction to Spark & Data processing

MUSIC & WEB

—

ARCHITECTURE & TECHNOLOGY OVERVIEW

MUSIC TRANSFORMATION

DIGITAL TRANSFORMATION

- Vinyl
- Cassette
- CDs
- MP3s & co

CONSUMPTION MODELS

- Ownership
    - Physical libraries
    - Digital libraries (local/remote)
- Access
    - Subscription
    - Pay As You Go

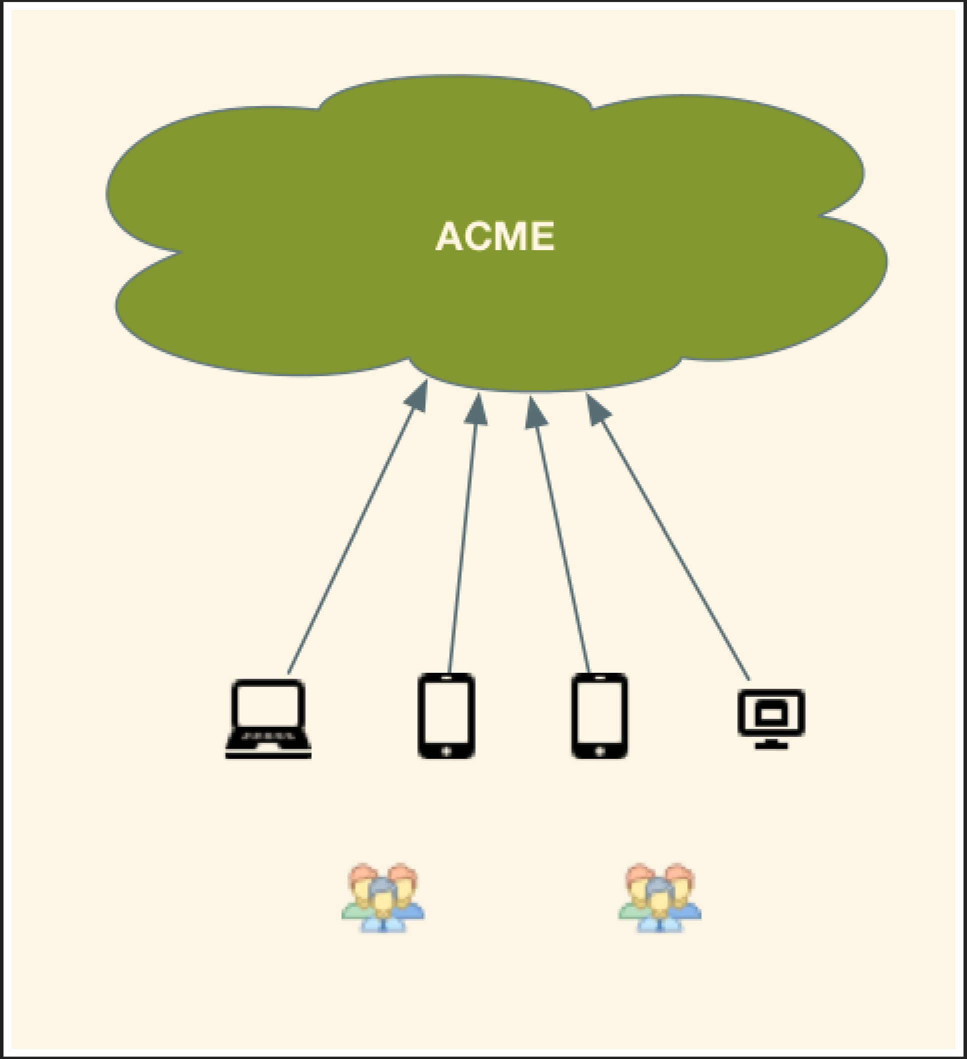
INTERNET

- Online stores
- Cloud libraries
- New services
  - Recommendation
  - Discovery
- Music is social

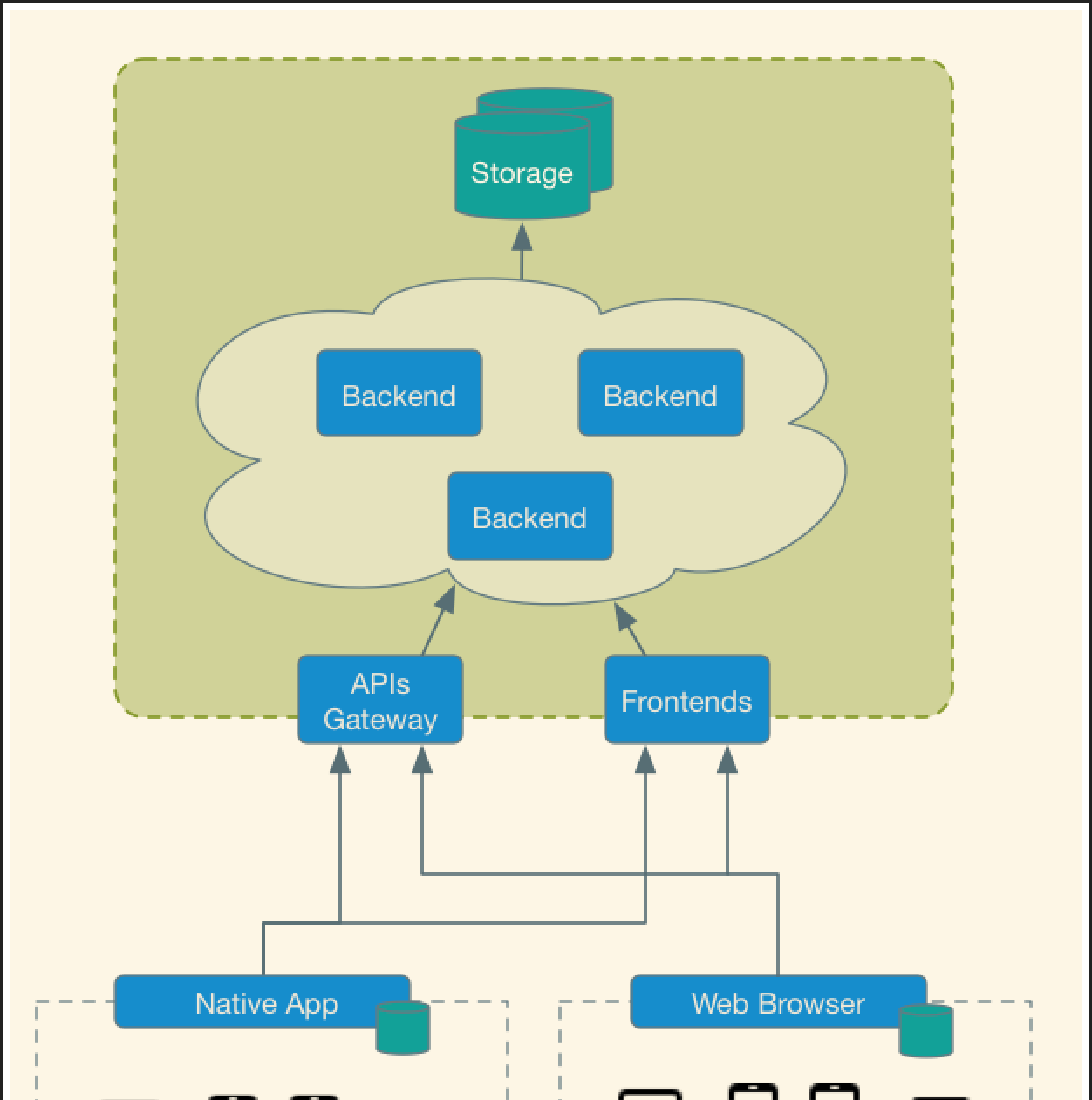
COMPONENTS OF A MUSIC SERVICE

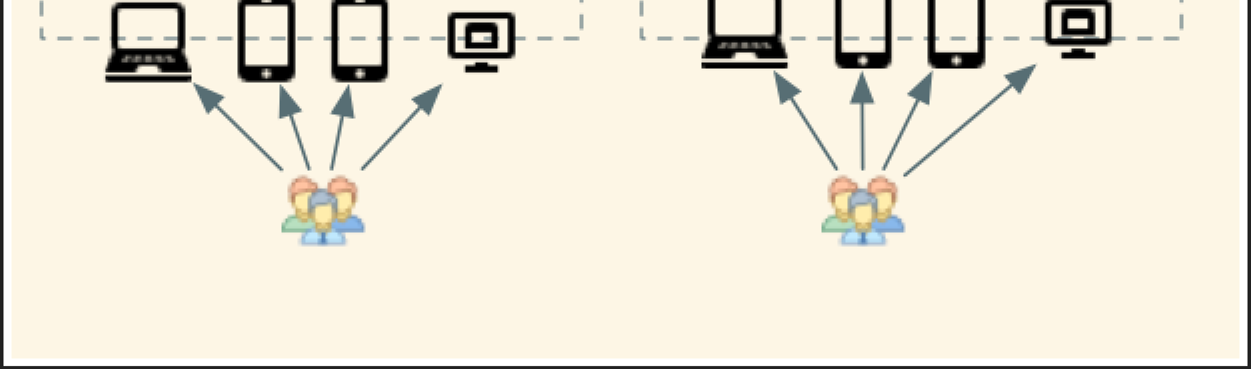
Thinking of building a Music service? :)

10,000 FEET VIEW



DIVE 1 - GENERIC ARCHITECTURE





API

An *Application Programming Interface* (API) is a set of subroutine definitions, protocols, and tools for building software and applications.

API

They are *everywhere*!

- OS (POSIX, Windows API, Cocoa, iOS, Android, ...)
- Software libraries (C++, Scala, Java, Python, Javascript, ...)
- Protocols, Remote APIs (JDBC, ...)
- Web API (SOAP, REST, ...)

API

API is not an implementation, only defines the *interface*

- Protocol
- Functions
- Data models of input/output objects

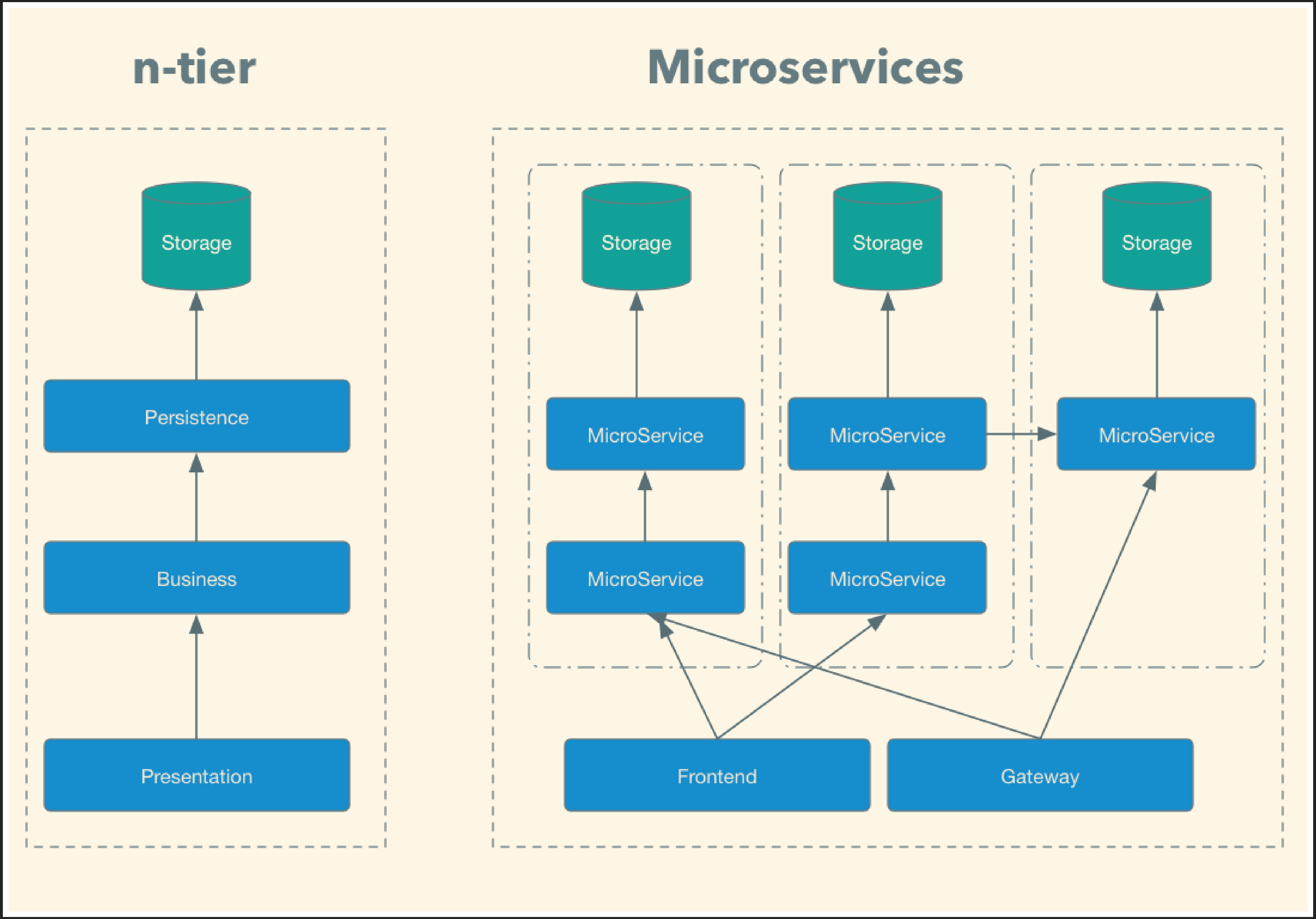
WEB SERVICE / WEB API

- Web Services
  - SOAP (Simple Object Access Protocol)
  - XML
- Web APIs
  - REST (Representational State Transfer)
  - JSON

WHY WEB API?

- Allow to easily build new applications
- Simplify development (API is fixed, known contract)
- Allow new services to be built, used (either internally or externally)
- Same approach than traditional software development
  - Components
  - Composing
  - Reusability
  - Testing
  - ...

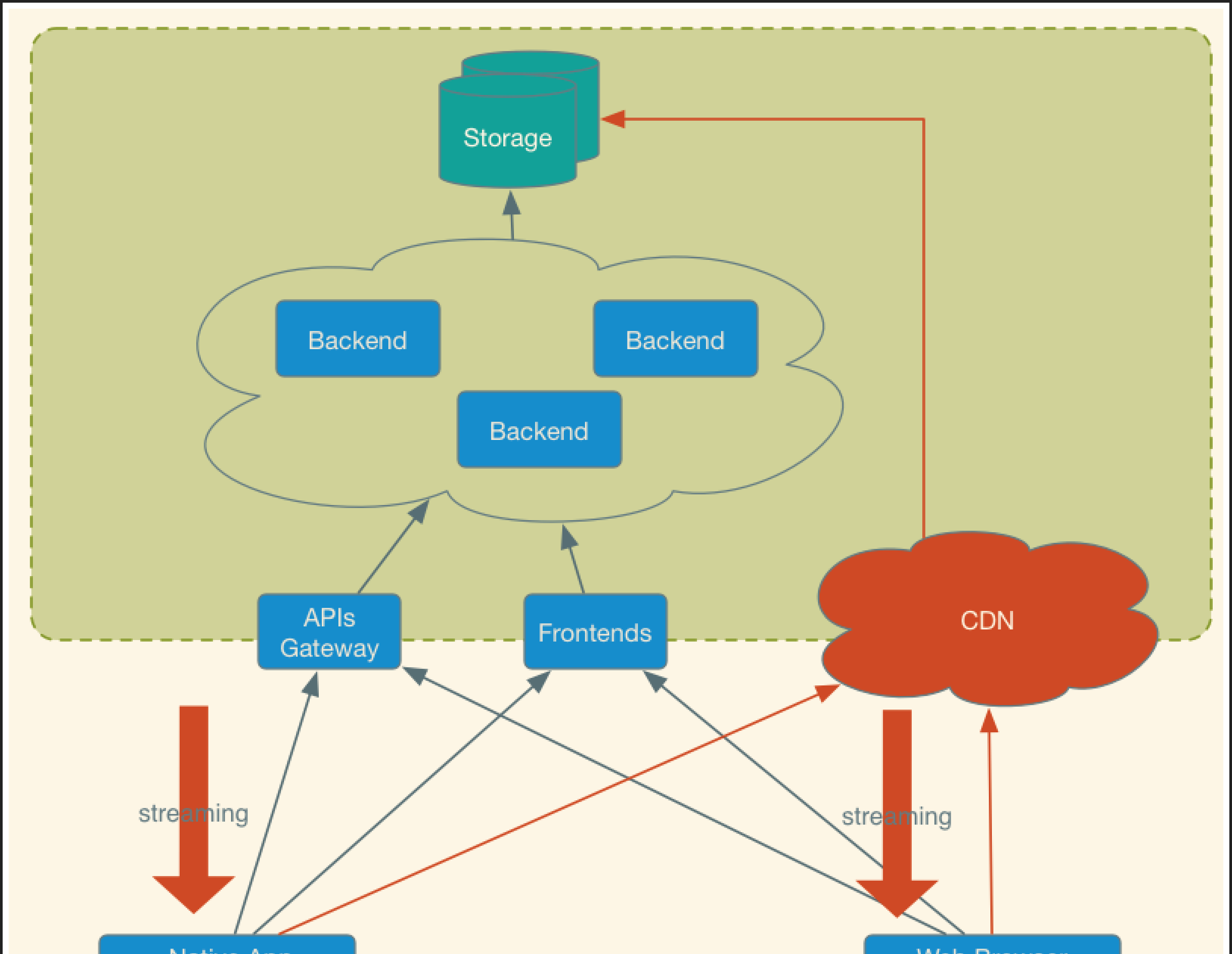
API - SYSTEM ARCHITECTURES

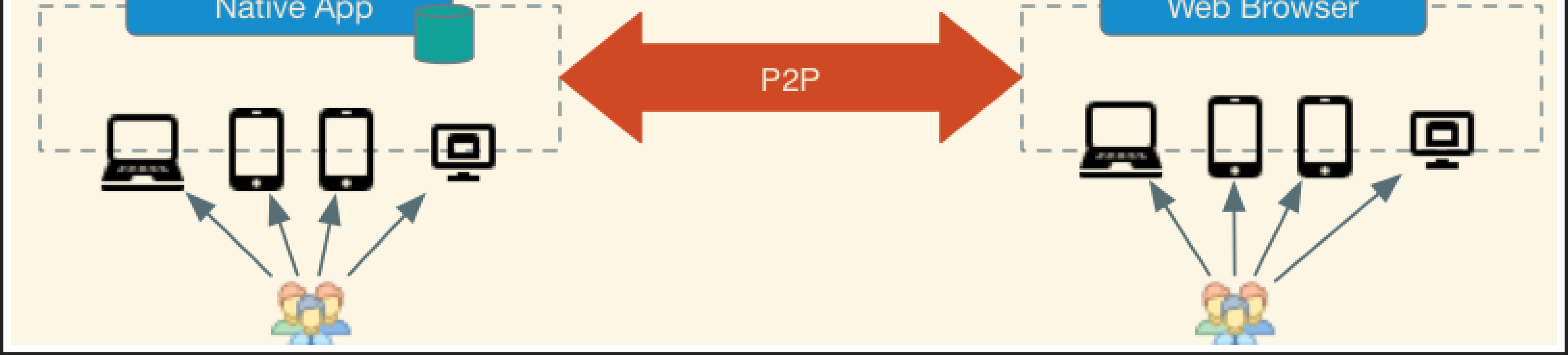

n-tier
Microservices
Storage
Persistence
Business
Presentation
Storage
Storage
Storage
MicroService
MicroService
MicroService
MicroService
MicroService
Frontend
Gateway

FRONTEND / BACKEND

- **Backend**
  - Business logic
  - API
- **Frontend**
  - Visualization
  - User interaction (UX)
- Decoupling eases:
  - development
  - testing
  - deployment

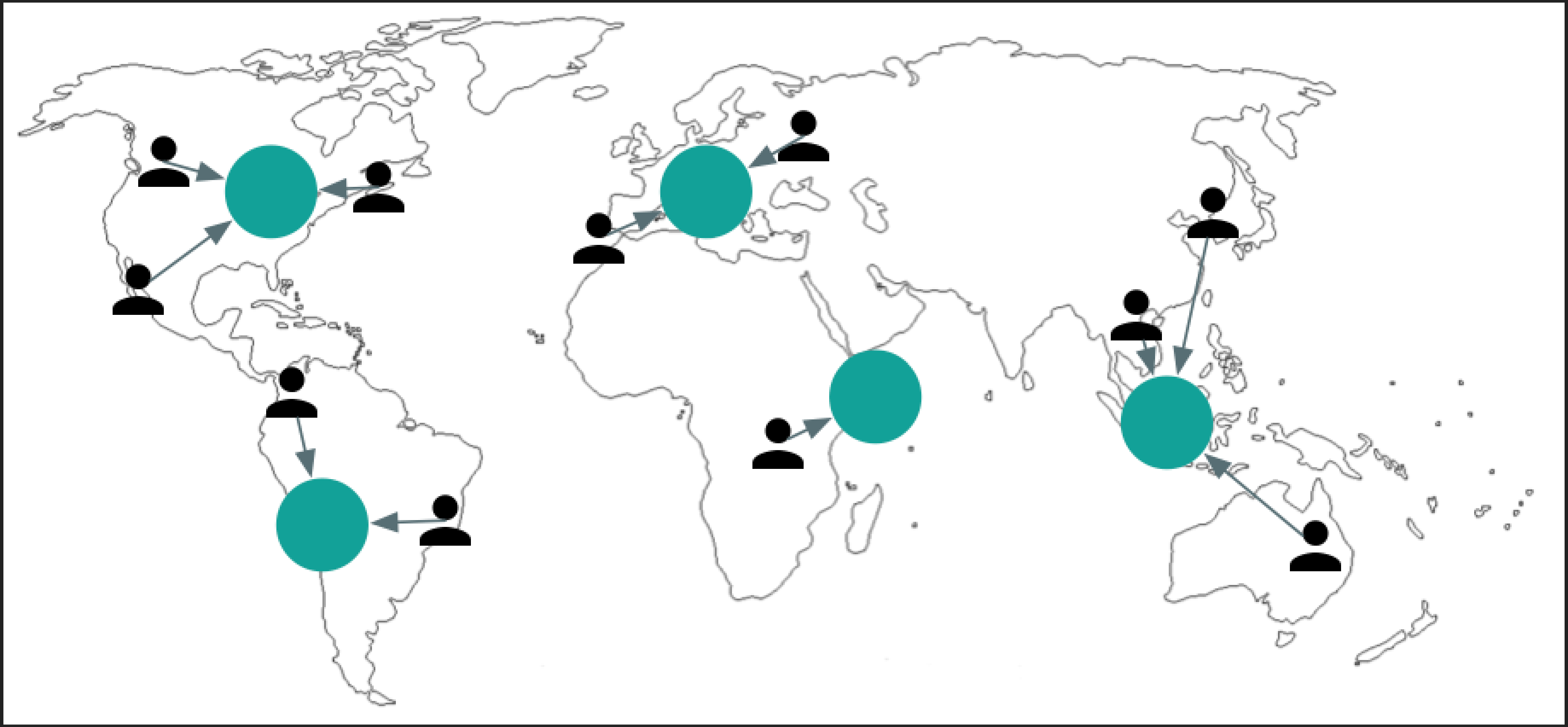
DIVE 2 - CONTENT DELIVERY




Native App
P2P
Web Browser

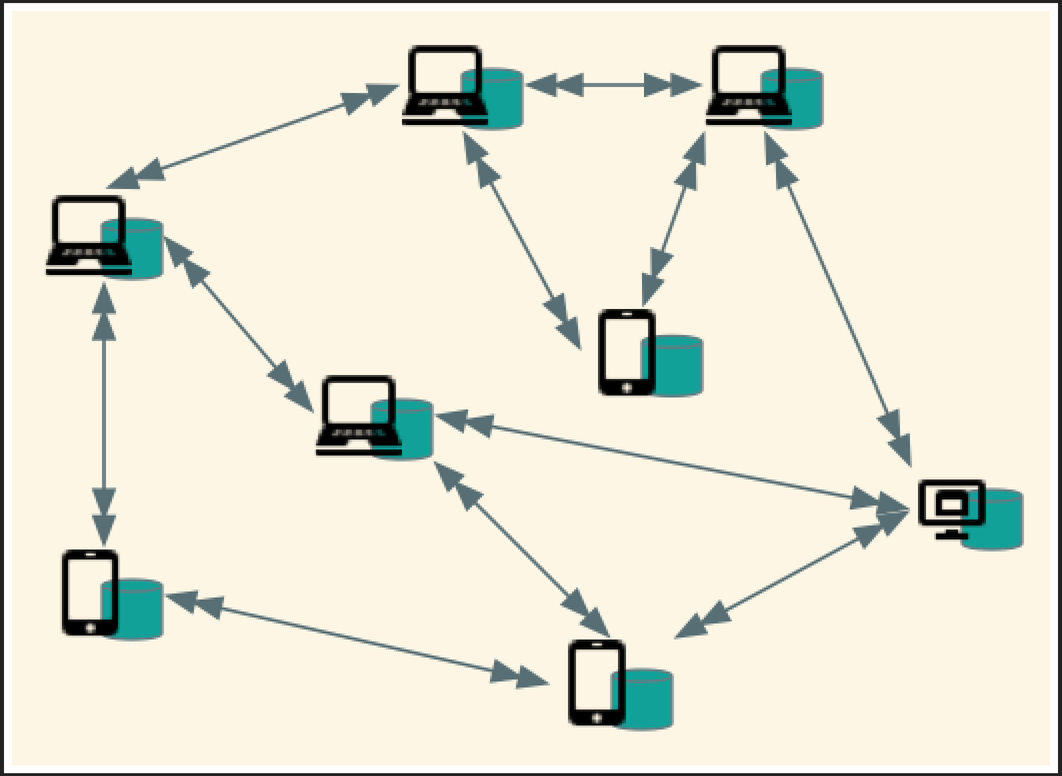
CDN

Content Distribution Network

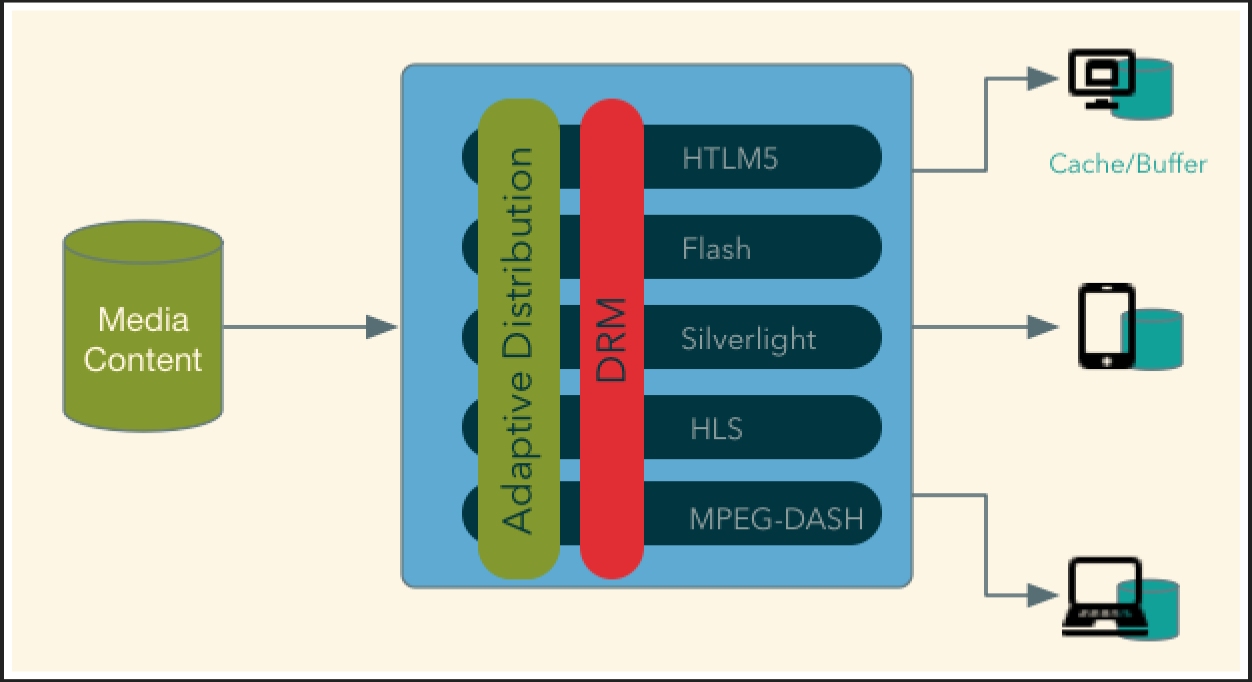


PEER 2 PEER

Decentralized network

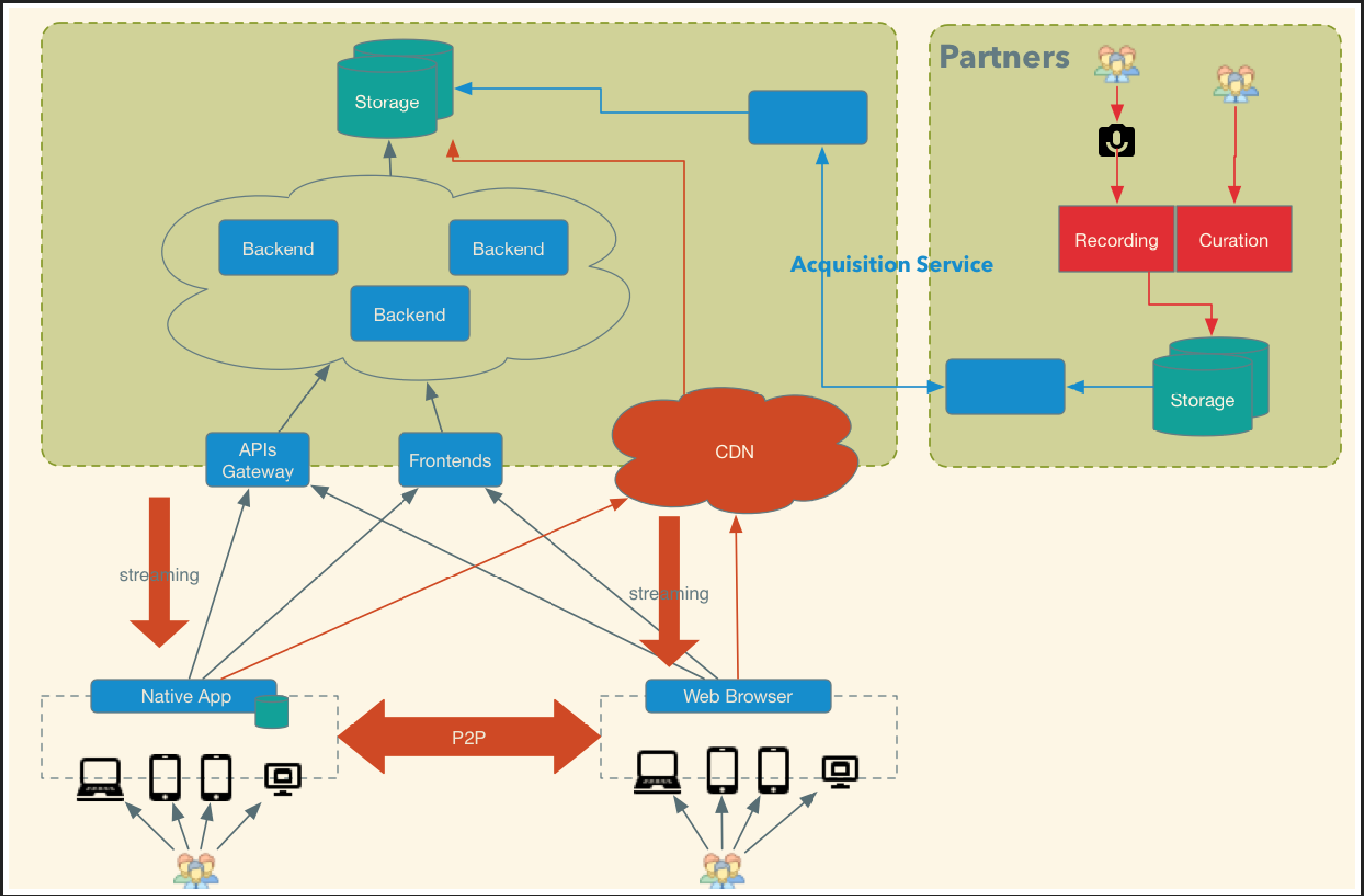


STREAMING

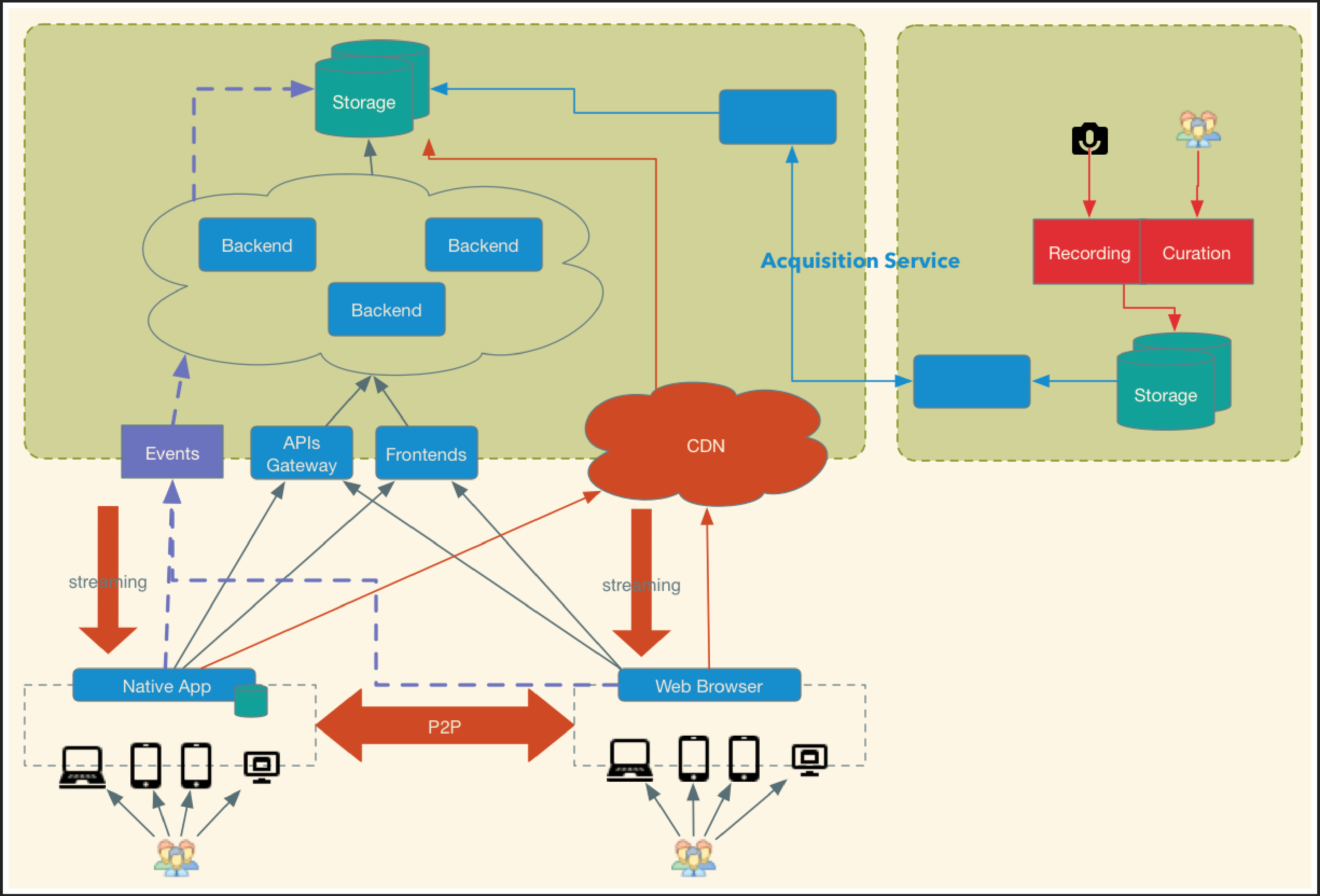


Media Content
Adaptive Distribution
DRM
HTLM5
Flash
Silverlight
HLS
MPEG-DASH
Cache/Buffer

DIVE 3 - PARTNERS



DIVE 4 - DATA & EVENTS COLLECTION



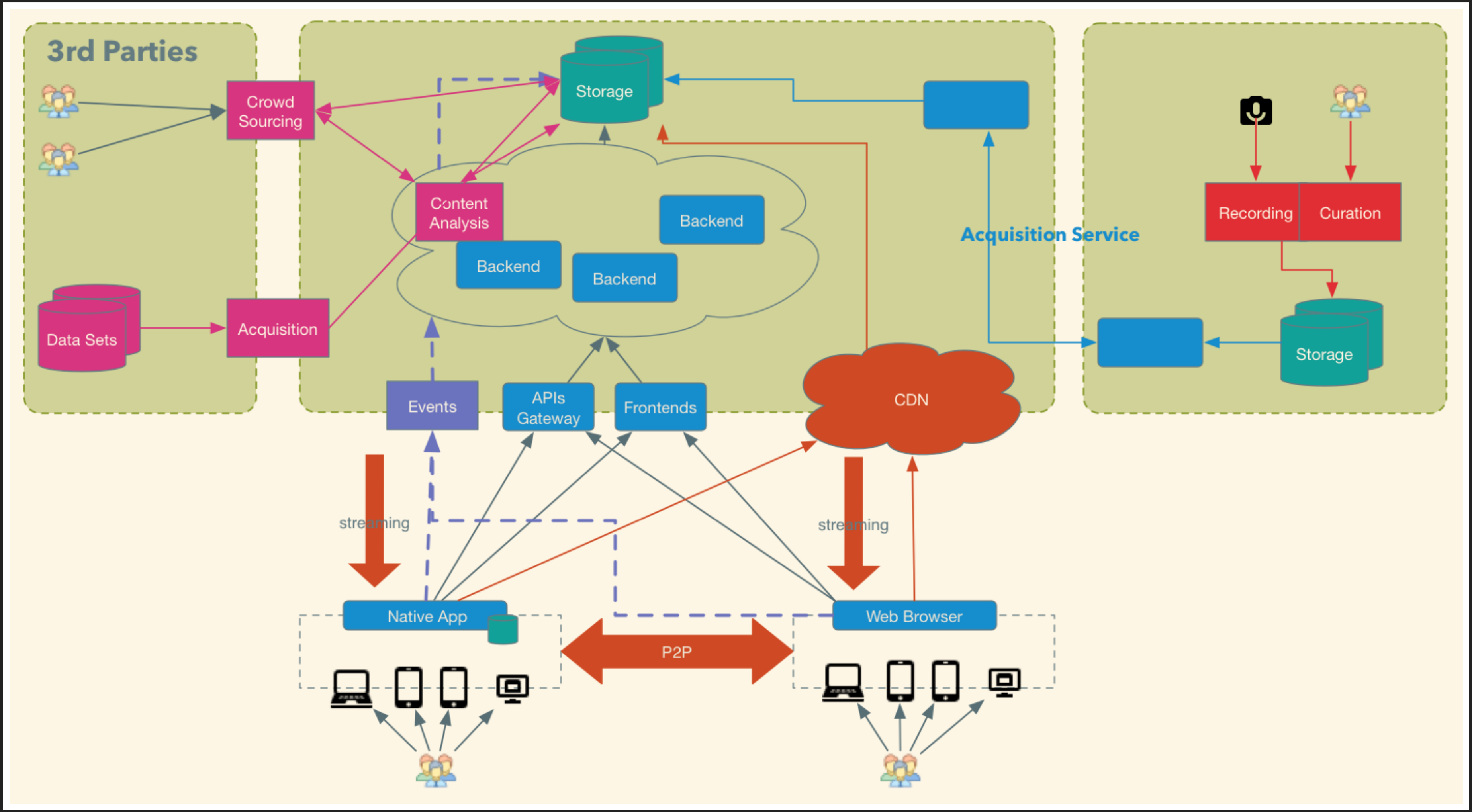
WHAT IS BEING COLLECTED ?

- Events generated by users
  - Click / Browsing activity
  - Listening activity
- Internal services
  - Performances, logs
- External services
  - Analytics, User engagement, tracking, ...
  - Social website monitoring (Facebook/Twitter feeds, ...)

WHY ?

- Marketing / Targeting
- Improve product quality
- Focus development on specific features
- Performance analysis / reliability
- Recommendation
- ...

DIVE 5 - CONTENT ANALYSIS & ENRICHMENT



CONTENT ANALYSIS & ENRICHMENT

- Metadata & content analysis
- Crowd sourcing
- Clustering & classification
- Fingerprinting
- Added content
- ...

DIVE 6 - INFRASTRUCTURE

Where do we run the different services?

WHERE?

- Physical Data Centers
  - On-premises
  - DC (owned or colocation)
- Cloud Infrastructure

DIFFERENT TYPES OF APPROACH TO INFRASTRUCTURE

IAAS

IaaS - Infrastructure As A Service

- Servers, Storage, Network, Operating System, ...

eg: Amazon EC2 & co, Windows Azure, Google Compute Engine, VmWare, OpenStack, ...

PAAS

> *PaaS - Platform As A Service*

- Managed databases, web servers, container solutions, ...

eg: AWS Elastic Beanstalk, AWS RDS, Heroku, Google App Engine, Cloud Foundry, ...

SAAS

SaaS - Software As A Service

- User facing software / consumption

eg: Google Apps, Office 365, Gmail, Dropbox, SalesForce, ...

WHO'S THERE?

QUESTIONS?

LAB !

LAB

Get the Jupyter notebooks

```
git clone https://github.com/glinmac/scimus-2016.git
```

Start Jupyter

```
cd scimus-2016
jupyter notebook
```