High Level Design

Microservices

Tanmay Kacker

Contents

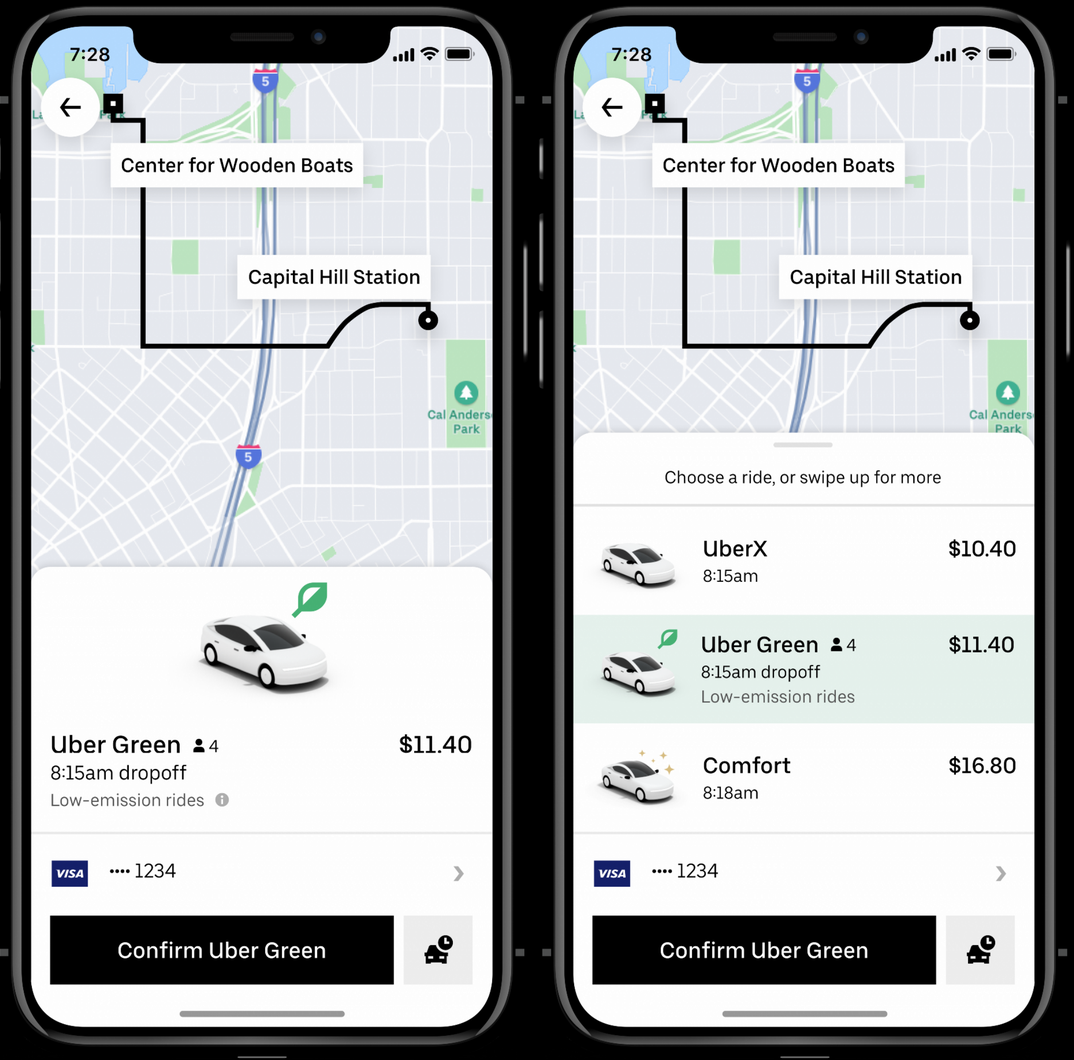
- Understanding a monolith with Uber
- Problems with monoliths
- Solution and microservices
- Case study 2 - Spotify

Case study

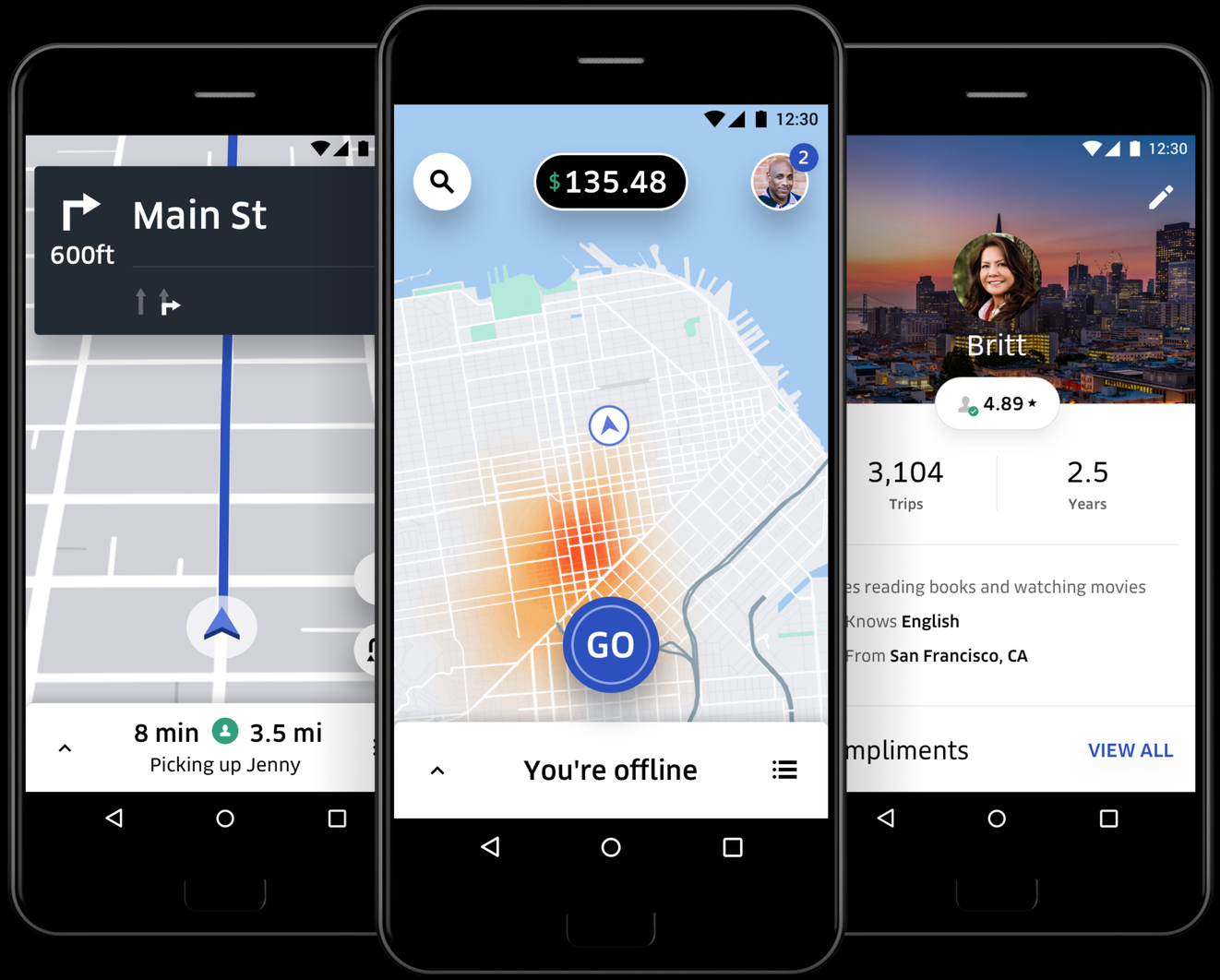
Uber

- Uber is primarily a ride hailing service
- Get a ride in minutes
- Or become a driver and earn money on your schedule.

Uber


7:28
Center for Wooden Boats
Capital Hill Station
Cal Anders Park
Uber Green 4
8:15am dropoff
Low-emission rides
$11.40
VISA •••• 1234
Confirm Uber Green
7:28
Center for Wooden Boats
Capital Hill Station
Cal Anders Park
Choose a ride, or swipe up for more
UberX
8:15am
$10.40
Uber Green 4
8:15am dropoff
Low-emission rides
$11.40
Comfort
8:18am
$16.80
VISA •••• 1234
Confirm Uber Green

Uber


Main St
600ft
8 min
3.5 mi
Picking up Jenny
12:30
$135.48
2
GO
You're offline
12:30
Britt
4.89
3,104
Trips
2.5
Years
es reading books and watching movies
Knows English
From San Francisco, CA
mpliments
VIEW ALL

Uber

Requirements

User management

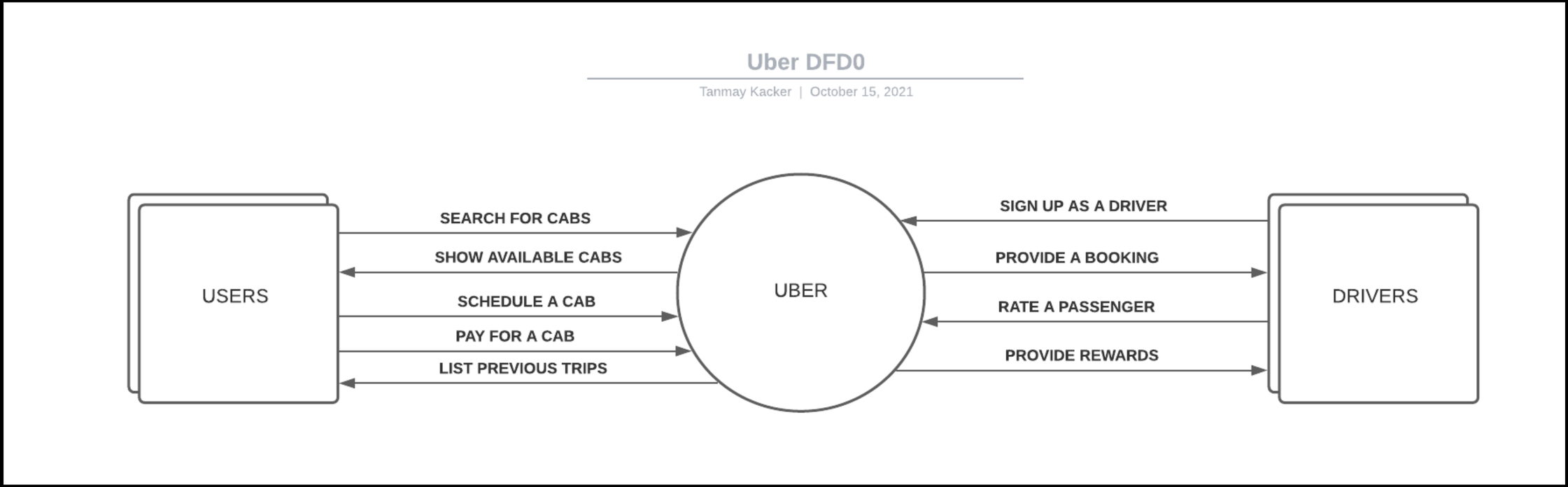
- Search for available cabs
- Schedule a cab
- Pay for a trip
- Get previous trips
- Rate a trip

Driver management

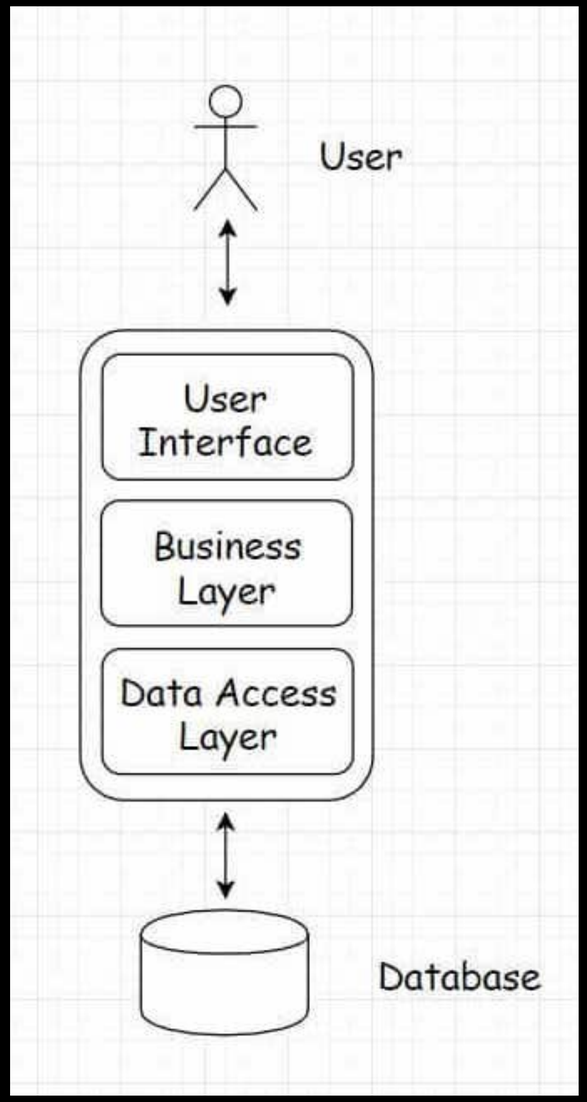
- Add a new driver
- Pay a driver
- Rate a passenger
- Reward a driver for ratings or number of trips

Monolithic System

Uber - Data flow



Monolithic system



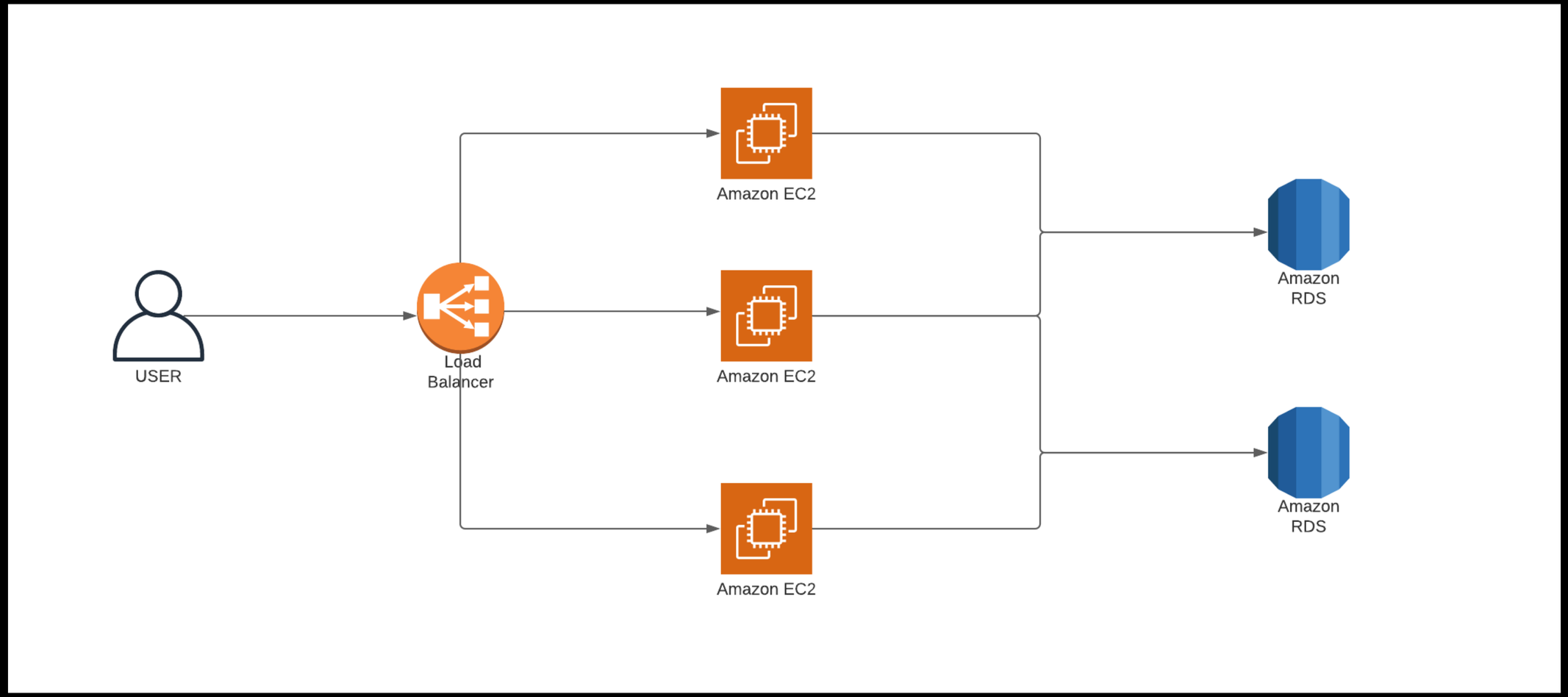
- Built as a single unit
- Deployed as a single executable

Monolithic System

Uber

Monolithic System

Uber



Monolithic System

Benefits

- Simple to develop
- Simple to deploy
- Simple to scale
- Easy to test

Uber
Numbers

The statistics

Facts and figures as of December 2018

91 million monthly active platform consumers

and 3.9 million drivers

10 billion trips

completed worldwide

63 countries

and 700+ cities

14 million trips

completed each day

There are over 22,000 employees at Uber

Monolithic System

Problems

- Joe Bloggs joins Uber as new software developer in the Driver management team.
- He is excited to to work at Uber. He starts exploring the codebase and is lost.
- All the code for passenger and driver management, reviews and payments team is present in the same repository.

Monolithic System

Problems - Complexity

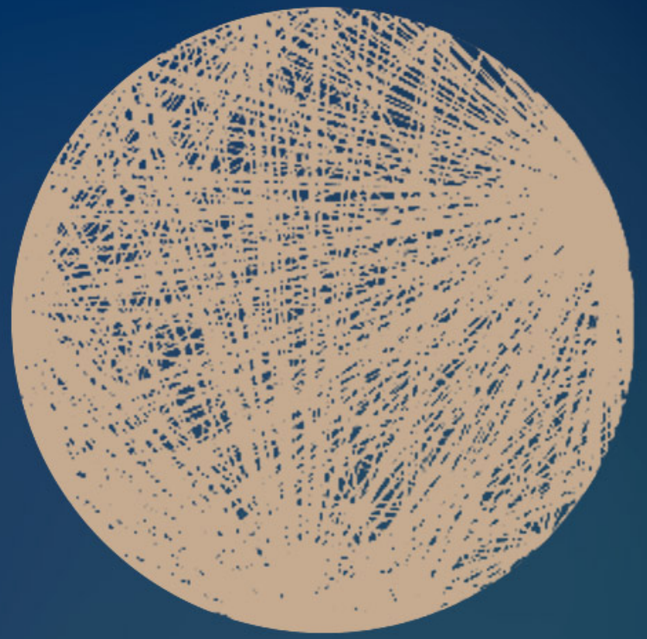
- Difficult to understand and modify
- Slow development
- Loss of modularity
- Declining code quality

Monolithic System

Problems - Complexity

Big Ball of Mud Dependency Graph

Dependencies
between components



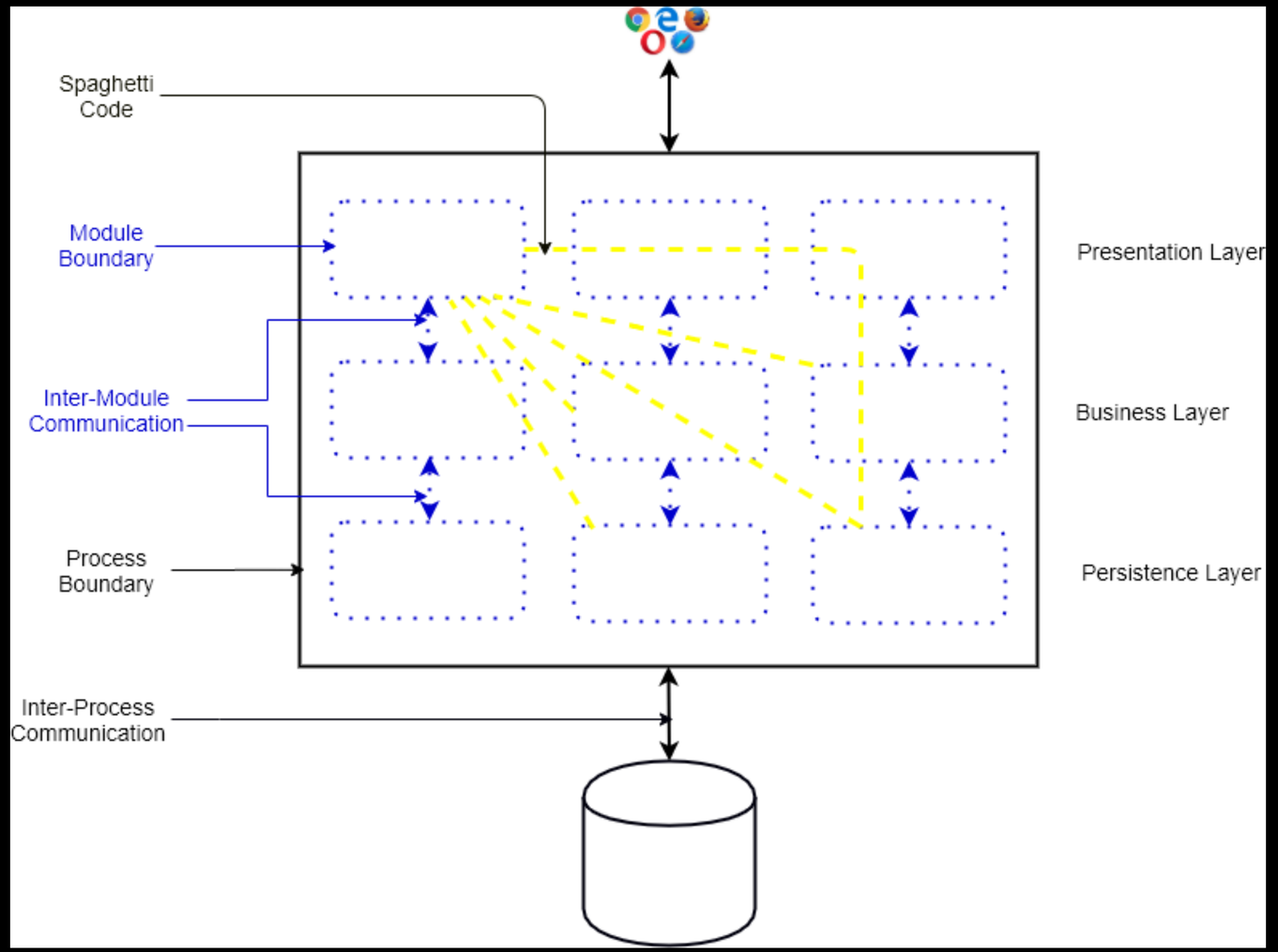
Monolithic System

Solutions - Complexity

- Modular monolith
- Packages and namespaces
- Git submodules

Monolithic System

Solutions - Modular Monolith



Monolithic System

Problems

- Joe Bloggs somehow figures out the codebase and gets started on his first task to send message to driver when he has achieved a trip.
- Joe Bloggs spends the afternoon working and is ready to get his first code merged. He is excited again.
- In order to deploy to small change, Joe Bloggs has to rebuild the complete the backend application and deploy the new version on all the application servers.
- Joe Bloggs is scared again.

Monolithic System

Problems - Deployment

- Continuous deployment is difficult
- In order to update one component you have to redeploy the entire application
- Background jobs need to be interrupted
- Redeployment becomes risky and discourages frequent deployment

Monolithic System

Solutions - Deployment

- Rolling vs batched deployment
- Complicated release cycle
- Searching for windows for deployment

Monolithic System

Problems

- Joe Bloggs is finally happy. His first piece of code is live on Uber.
- Joe Bloggs moves on to his next task. He has improve the logic behind searching of drivers nearby passengers. He has to filter out drivers who have an expired driving license.
- This is a critical task since this API is called by all the visitors on the application and used internally as well.
- Sadly, there was a memory leak in the code and all the servers are affected.
- A small bug caused a major outage for all of Uber. Poor Joe.

Monolithic System

Problems - Blast radius

- Due to how interconnected and interdependent the components are, even a minor issue leads to the failure of the entire application.
- Faults are global and not local

Monolithic System

Solutions - Blast radius

- Staging environments and extensive testing
- Complicated ways of segregating components
- Example - multiple environments and URL based routing

Monolithic System

Problems

- Joe Bloggs and Uber finally recovered from the incident. Fun first month for Joe.
- Joe Bloggs has to now look into the memory issue which caused the downtime last time.
- Joe identifies that the current machines are optimised for CPU and in order to avoid the issue again they should move to memory optimised machines.
- Apparently the payments team had requested for CPU optimised machines.

Monolithic System

Problems - Scaling

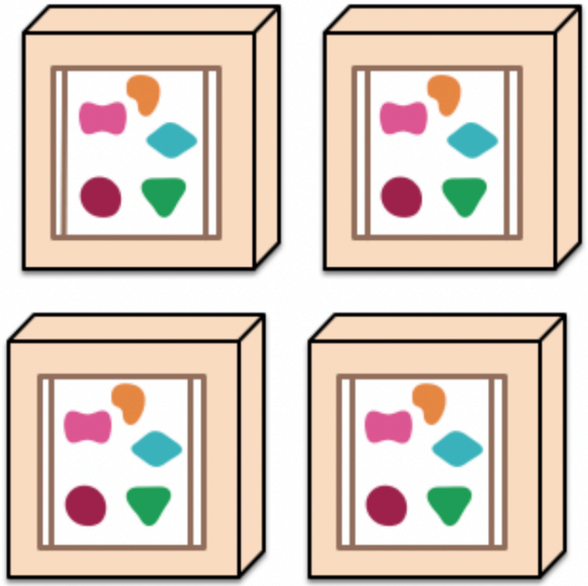
- A monolith only scales in one dimension
- Each copy of application instance will access all of the data, which makes caching less effective and increases memory consumption and I/O traffic
- Components cannot scale independently for resource requirements
- Memory intensive vs CPU intensive

Monolithic System

Problems - Scaling

A monolithic application puts all its functionality into a single process...

... and scales by replicating the monolith on multiple servers



Monolithic System

Solutions - Scaling

- Automatic and dynamic scaling
- Back to multiple environments
- Each environment can cater to resource requirements

Monolithic System

Problems

- Complexity of codebase
- Loss in modularity
- Risky deployment
- Speed of development
- Large blast radius
- Lack of scalability
- Loss of flexibility with technology stack

Further reading

Monolithic architecture

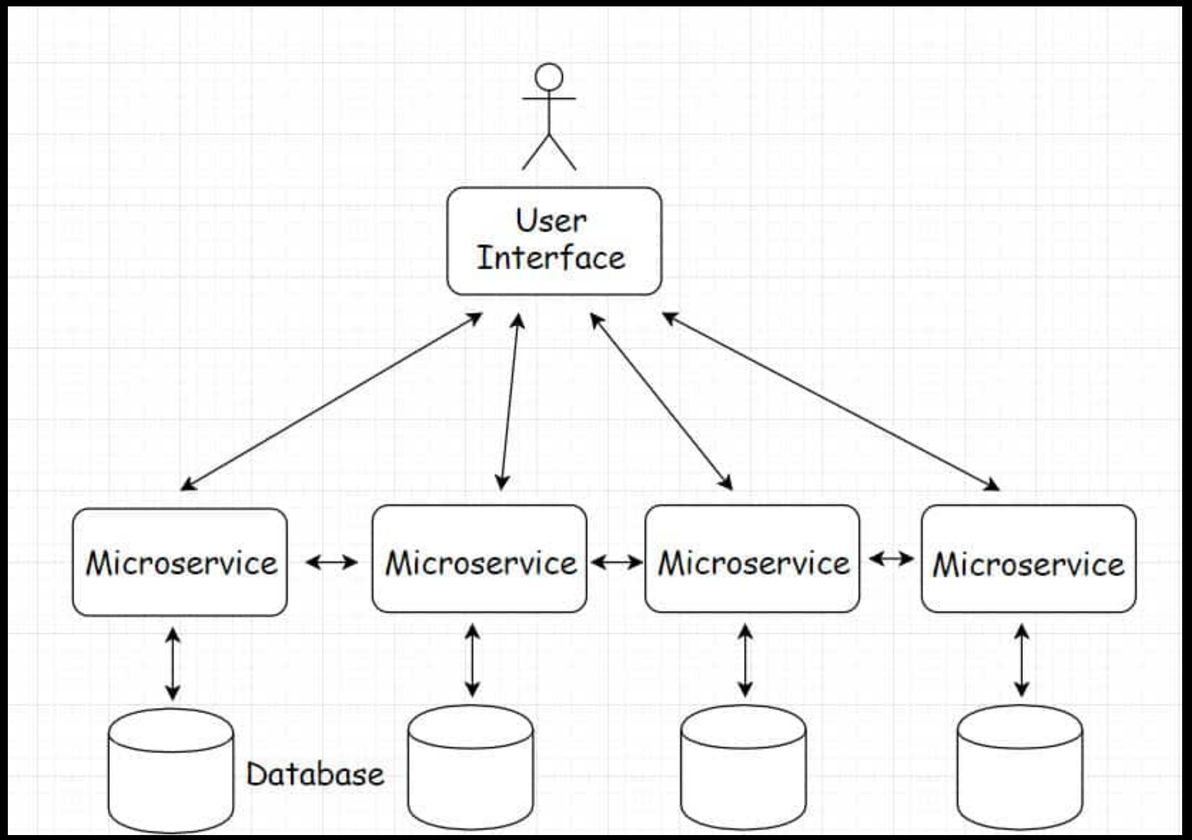
- What is a data flow diagram
- Scale cube
- Big Ball of Mud
- Spaghetti Code
- Modular monolith

References

Monolithic architecture

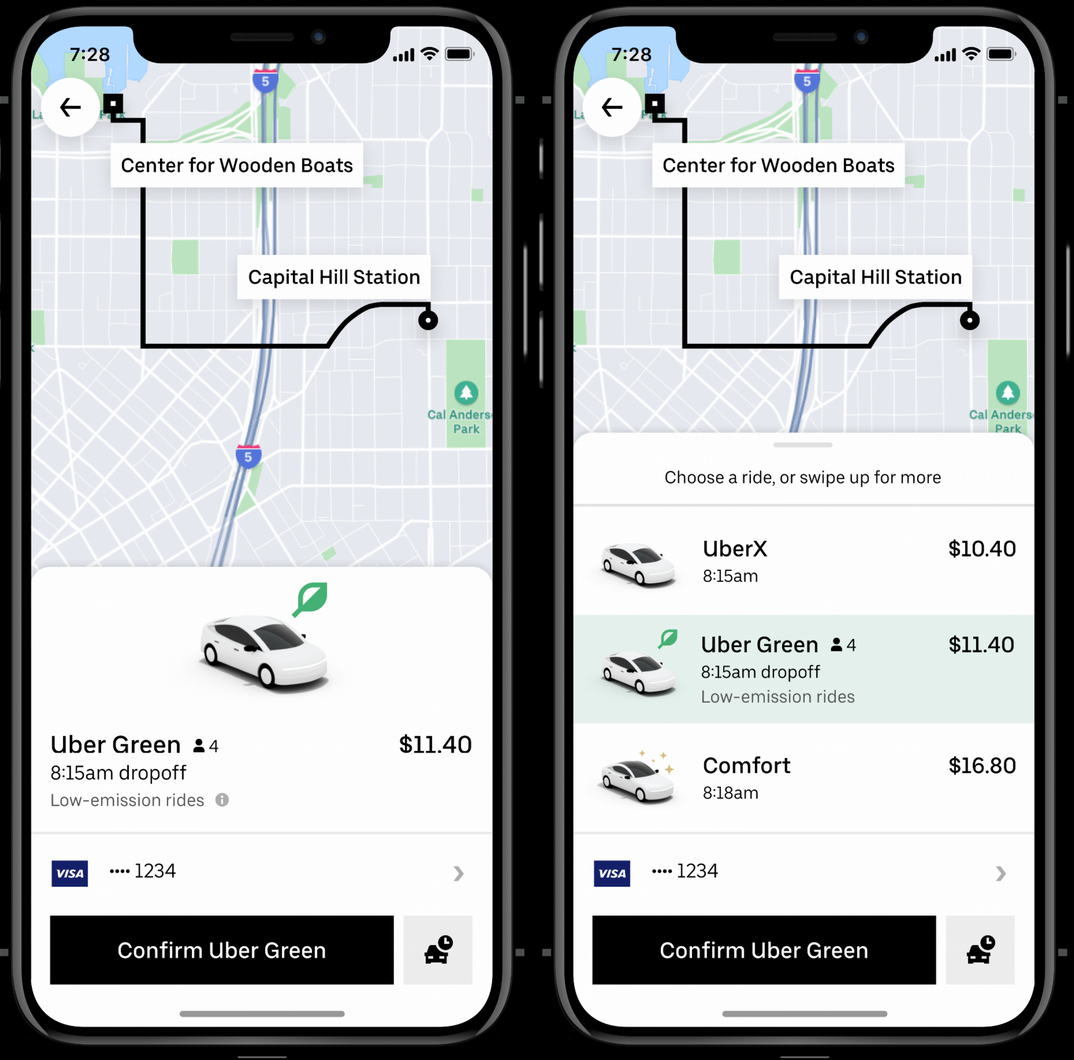
- Breaking down the monolith
- Monolithic Architecture
- Uber case study

Microservices

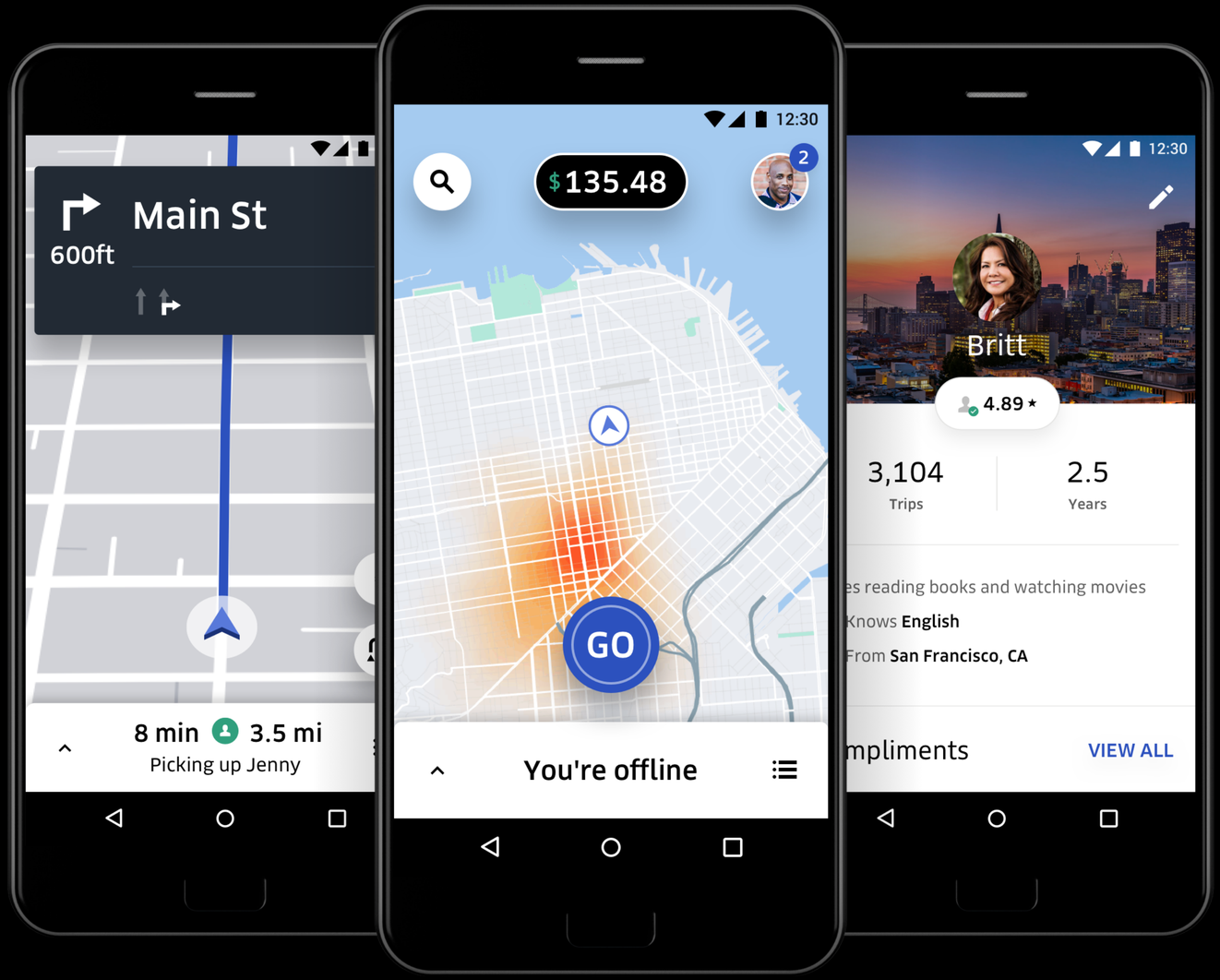


Single application as a suite of small services

Uber


7:28
Center for Wooden Boats
Capital Hill Station
Cal Anders Park
Uber Green 4
8:15am dropoff
Low-emission rides
$11.40
VISA •••• 1234
Confirm Uber Green
7:28
Center for Wooden Boats
Capital Hill Station
Cal Anders Park
Choose a ride, or swipe up for more
UberX
8:15am
$10.40
Uber Green 4
8:15am dropoff
Low-emission rides
$11.40
Comfort
8:18am
$16.80
VISA •••• 1234
Confirm Uber Green

Uber


Main St
600ft
8 min
3.5 mi
Picking up Jenny
12:30
$135.48
2
GO
You're offline
12:30
Britt
4.89
3,104
Trips
2.5
Years
es reading books and watching movies
Knows English
From San Francisco, CA
mpliments
VIEW ALL

Uber

Requirements

User management

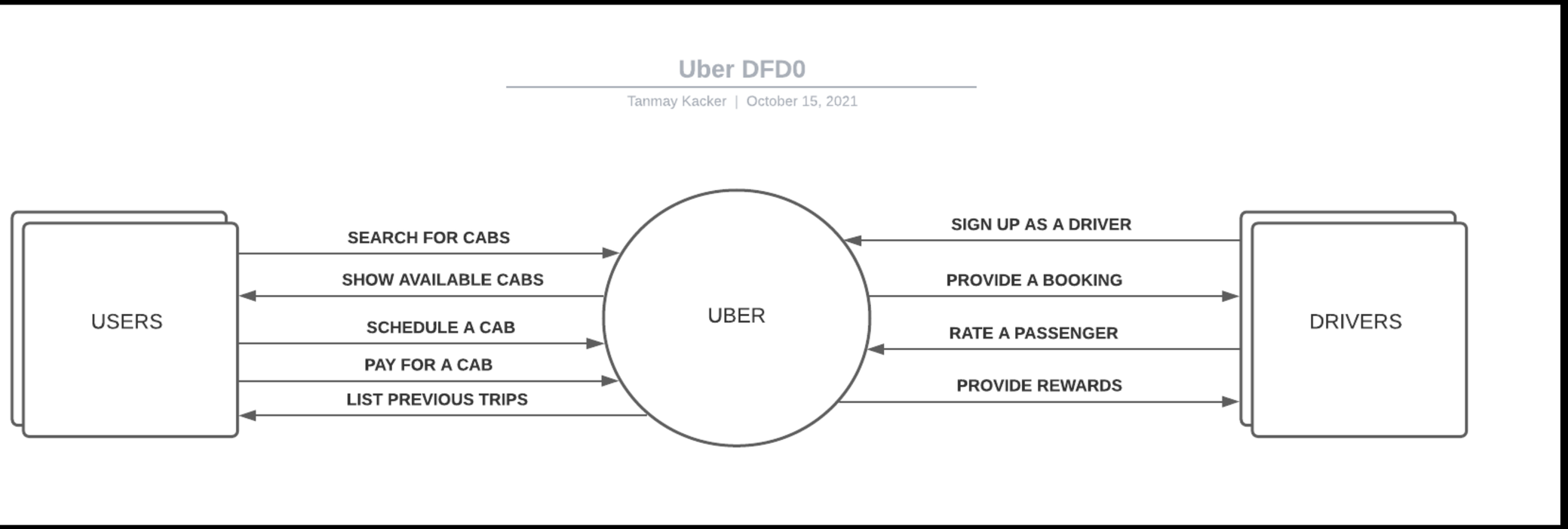
- Search for available cabs
- Schedule a cab
- Pay for a trip
- Get previous trips
- Rate a trip

Driver management

- Add a new driver
- Pay a driver
- Rate a passenger
- Reward a driver for ratings or number of trips

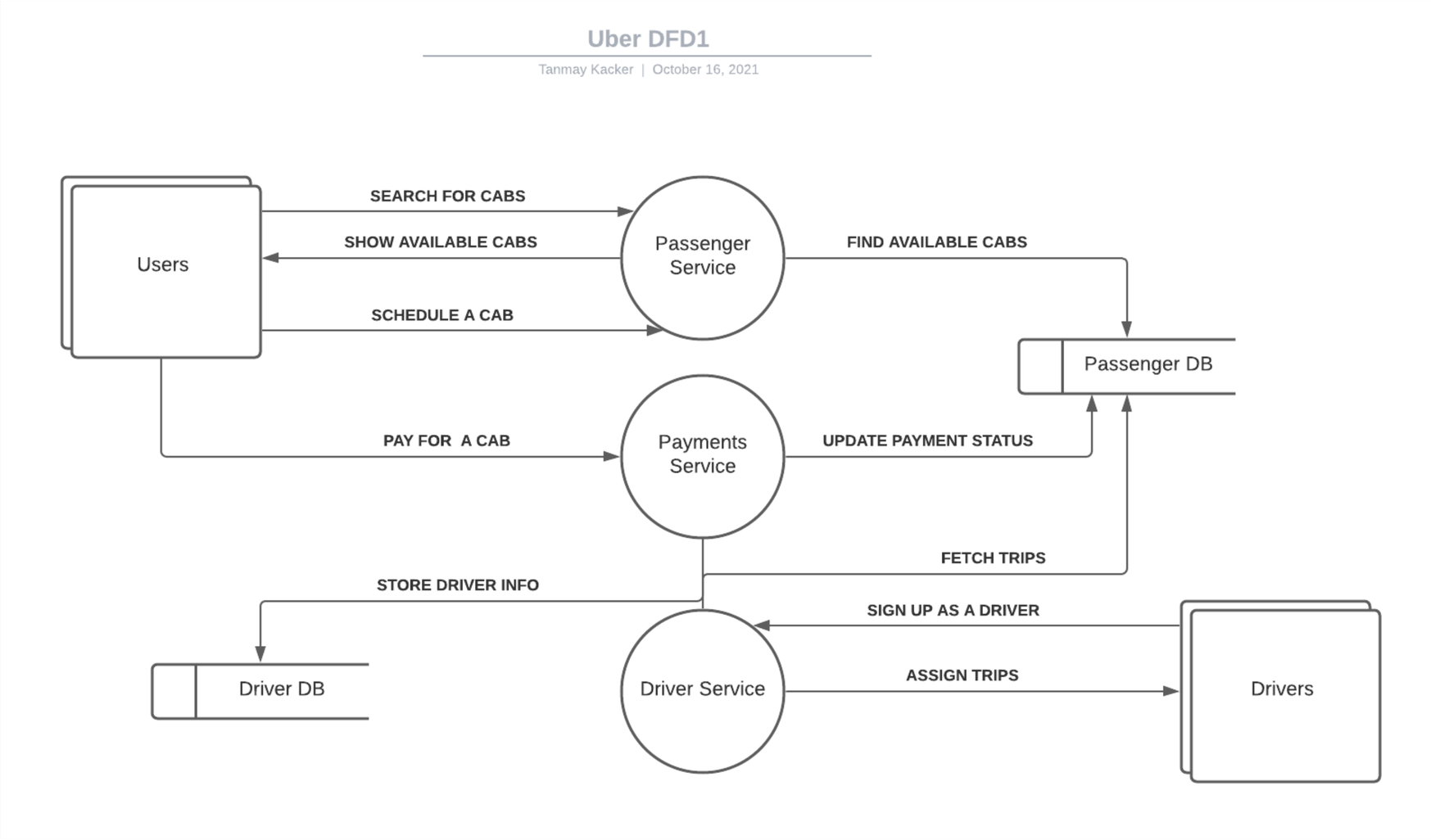
Uber

Data flow



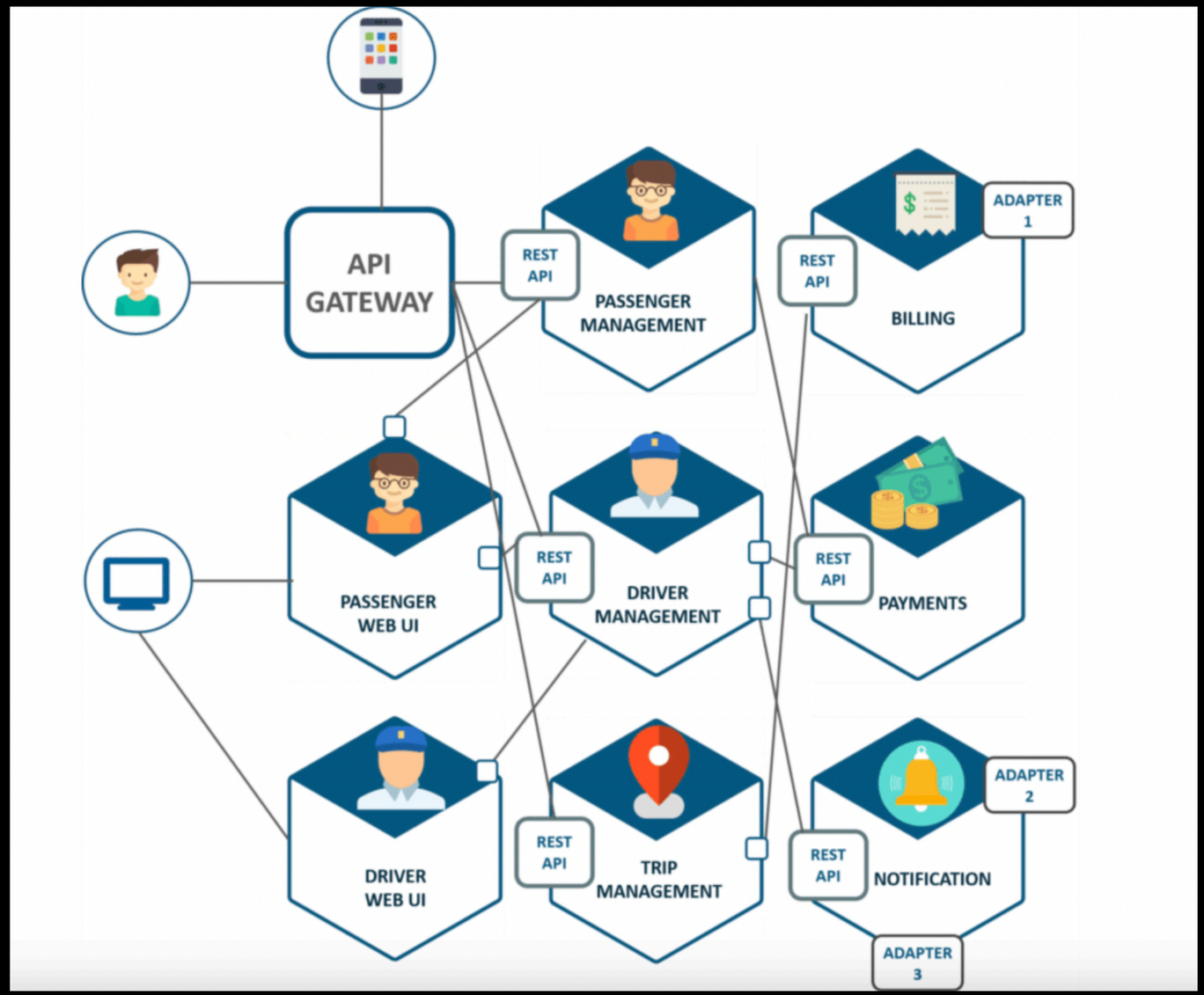
Monolithic System

Uber - Data flow



Microservices

Uber



Microservices

Problems of a monolith

- Complexity of codebase
- Loss in modularity
- Risky deployment
- Speed of development
- Large blast radius
- Lack of scalability
- Loss of flexibility with technology stack

Microservices

Solutions - Maintainability

- Each service is relatively small and so is easier to understand and change
- Easier for a developer to understand
- Enables you to organize the development effort around multiple, autonomous teams.
- Each team owns and is responsible for one or more services.
- Each team can develop, test, deploy and scale their services independently of all of the other teams.
- Improved maintainability

Microservices

Solutions - Deployment

- Services can be deployed independently
- Application start up time is less
- Speeds up deployment

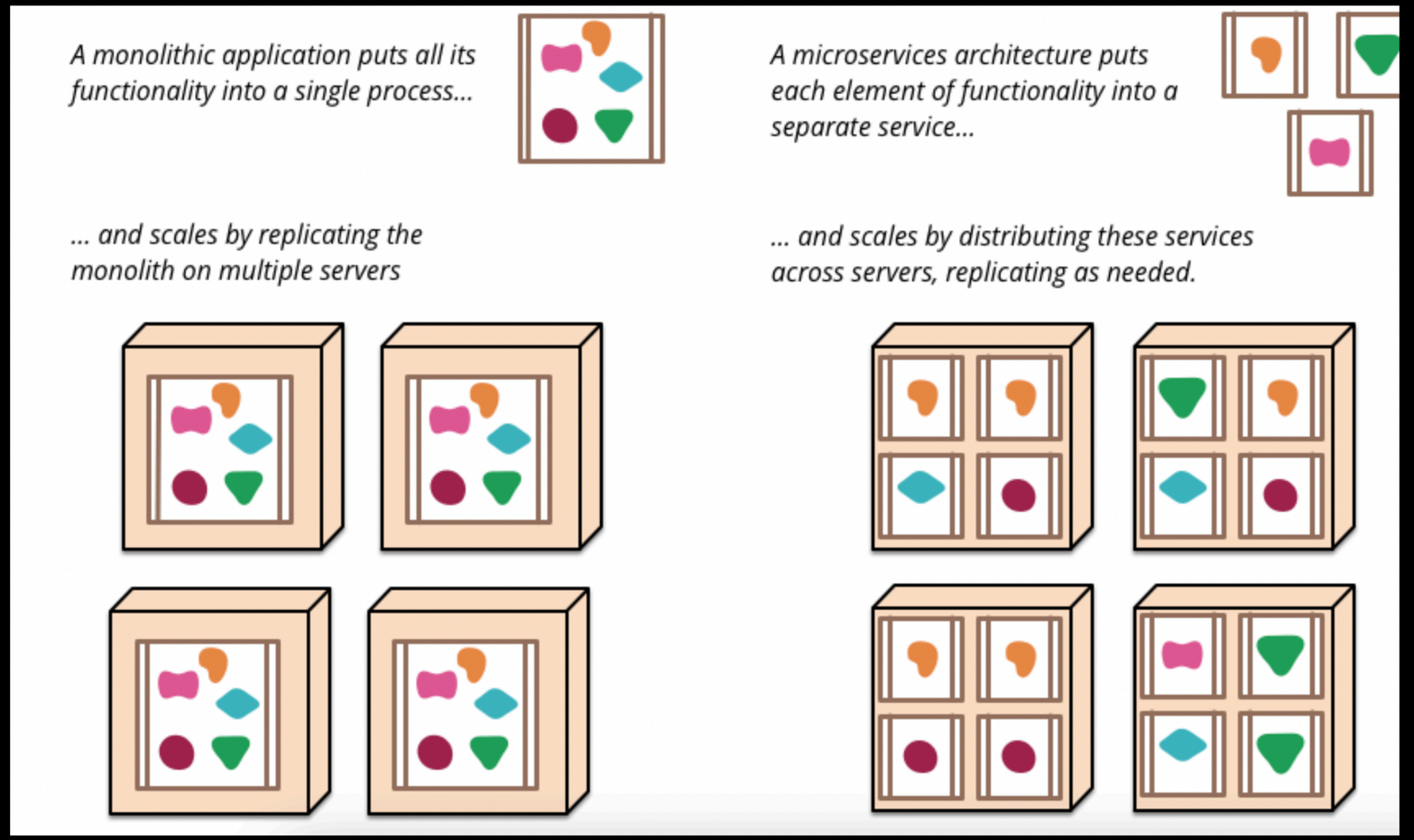
Microservices

Solutions - Blast radius

- Improved fault isolation
- If a service is affected all other services can continue to function unaffected

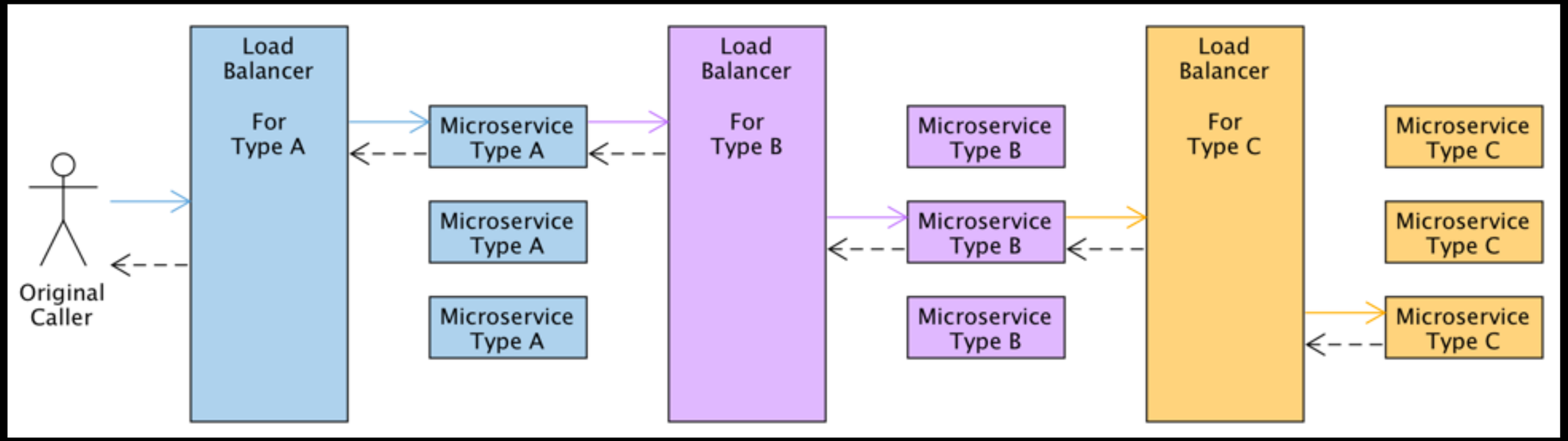
Microservices

Solutions - Scalability



Microservices

Problems



Multiple network hops

Microservices

Problems

- Greater effort
- Testing the interactions between services is more difficult
- Requires coordination between teams
- Development across services can be difficult

Microservices

Breaking down applications

- Single responsibility principle
- Each service should be responsible for one business capability

Further reading

Microservices

- Decomposition by domain
- Decomposition by business capability
- Monolith First
- API gateway
- Netflix API gateway
- Client-side service discovery
- Server-side service discovery

References

Microservices

- Microservices - I
- Monolithic Architecture
- Uber case study

System Design

- Spotify is a digital music, podcast, and video service.
- Spotify gives you access to millions of songs and other content from creators all over the world.

Spotify®


Devices
App Finder
YOUR MUSIC
Songs
Albums
Artists
Local Files
Morning Commute
Make it to work the right
Your Favorite Coffee
Ready for the Day
This fun, upbeat playlist
ill help get you up and
dy to start your day
Feelin Good
Feel good with this positively timeless Soul playlist
On Wheels To Work
Take your bike to work, breathe in the fresh air, and listen to these tunes
Spotify
SPOTLIGHT
ARTIST · VIEW PROFILE
Foster The People
FOLLOWERS
1,079,828
FOLLOWING
PAUSE
FOLLOWING
OVERVIEW
RELATED ARTISTS
BIOGRAPHY
POPULAR
1 Pumped up Kicks 92,077,824
2 Helena Beat 34,493,970
3 Houdini 28,992,336
4 Don't Stop (Color on the Walls) 23,494,826
Best Friend 8,897,687
SHOW 5 MORE
ALBUMS
RELATED ARTISTS
Coldplay
The Killers
Mark Foster
Elite Gymnastics
RELATED MUSIC
Best Friend
Foster The People
Sex
The 1975
Sleepyhead
Passion Pit
The Mother We ...
CHVRCHES
Contact High
Architecture In Helsinki
The Struggle
Grizfolk
Miss You
Foster The People
Best Friend
Foster The People
1:57
4:27
Search
Browse
Radio
Your Music
Follow
Sara
FOSTER THE PEOPLE SUPERMODEL
FOSTERTHEPEOPLE
09:41
9:41 AM
100%
Foster The People
FOLLOWING
1 079 908 FOLLOWERS
SHUFFLE PLAY
POPULAR
1 Pumped up Kicks
92 077 824
2 Helena Beat
34 493 970
Best Friend
Foster The People
PLAYING FROM ARTIST
Calvin Harris
CALVIN
HARRIS
BLAME
FEAT. JOHN NEWMAN
Blame
Calvin Harris
0:54
3:34

Spotify

Numbers

Subscribers:

96 million

Monthly active users:

207 million

Tracks:

40+ million

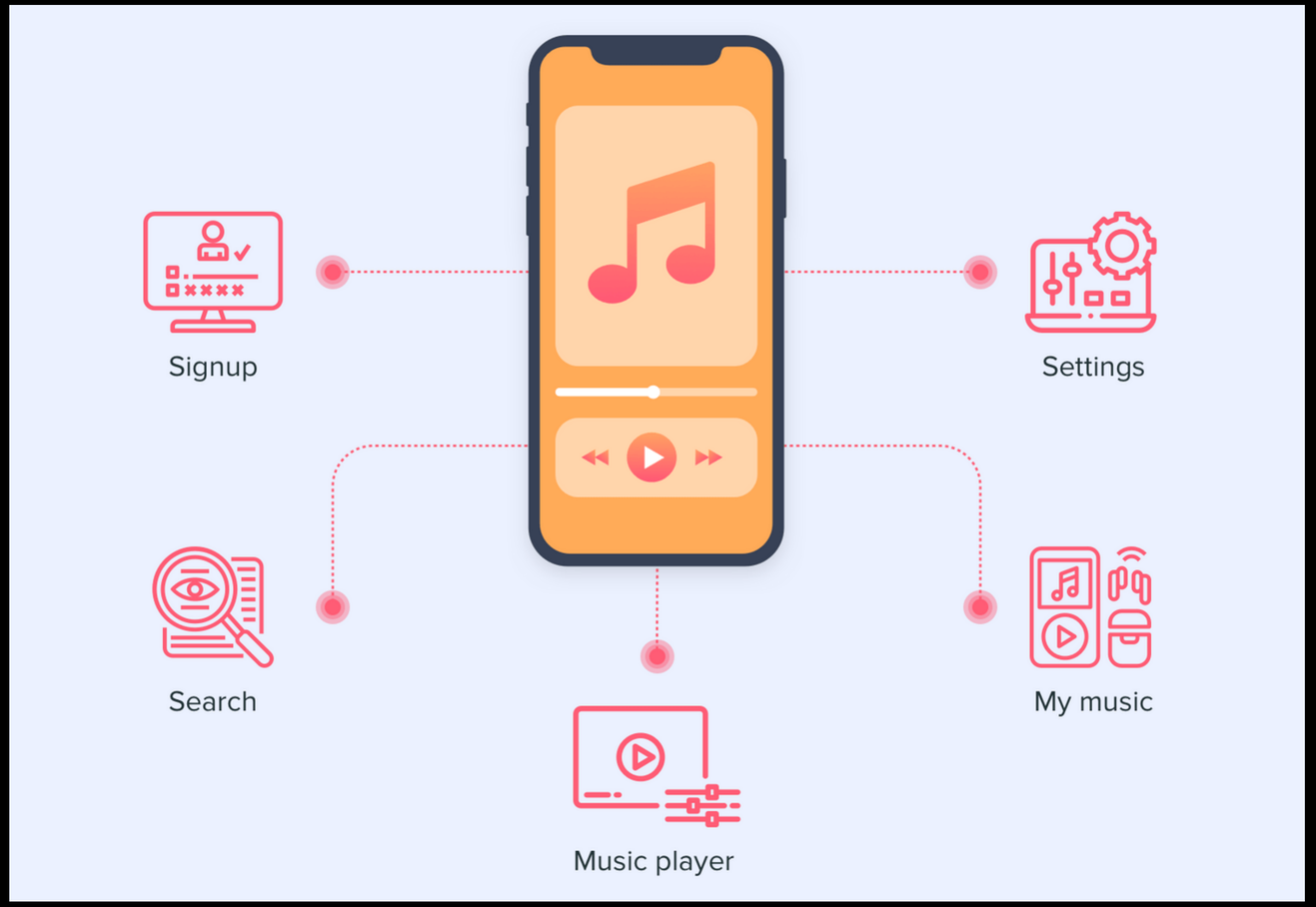
Available in:

78 markets

Source https://investors.spotify.com/home/default.aspx

Spotify

Requirements



Spotify

Scope

- Requirements
- Data flow
- Artist to Spotify architecture
- Spotify to user architecture
- Event delivery system

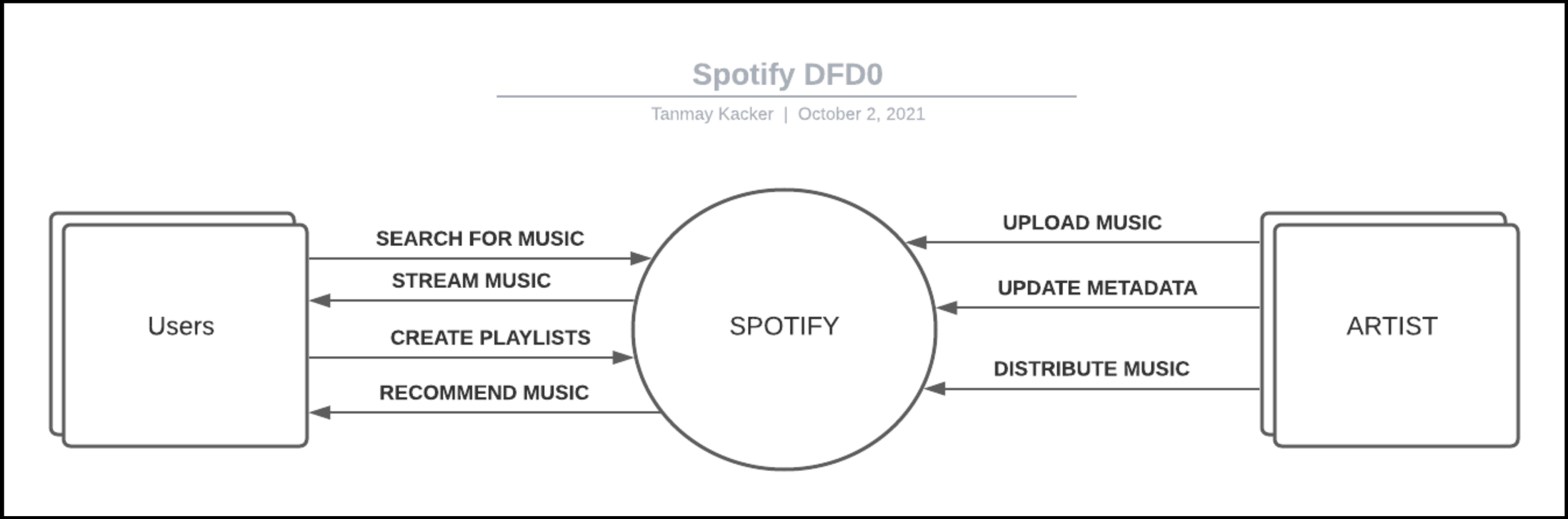
Spotify

Requirements

- Search for music
- Listen to music
- Add music to playlist
- Upload music
- Distribute music

Spotify

Data Flow (DFD0)



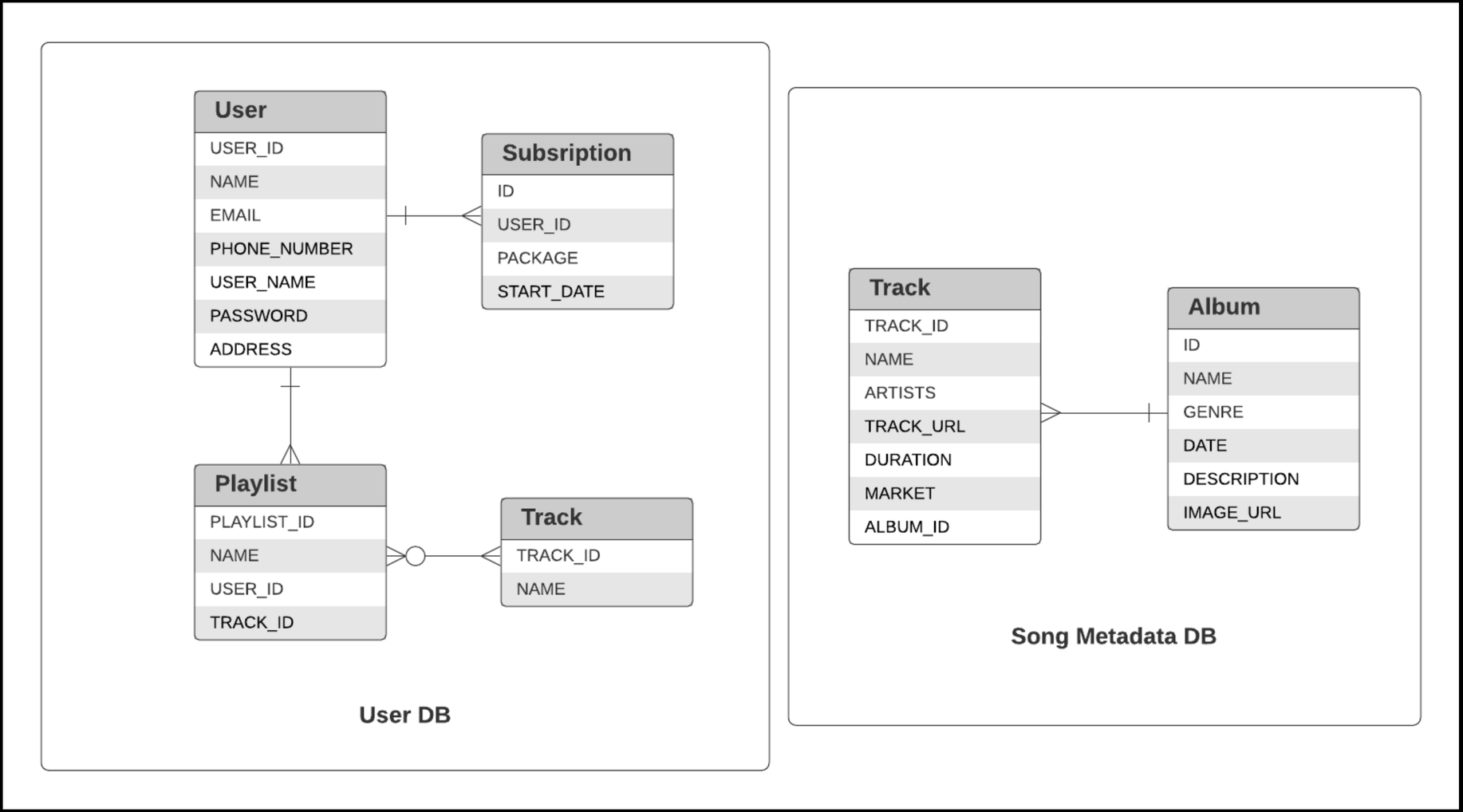
Spotify

Sample Microservices

- Account service (username, password)
- Search service (song name, album name, lyrics)
- Song service (song ID)
- Playlist service (song ID, playlist ID)
- Publisher service (genre, artists, market, labels, name, tracks, restrictions)

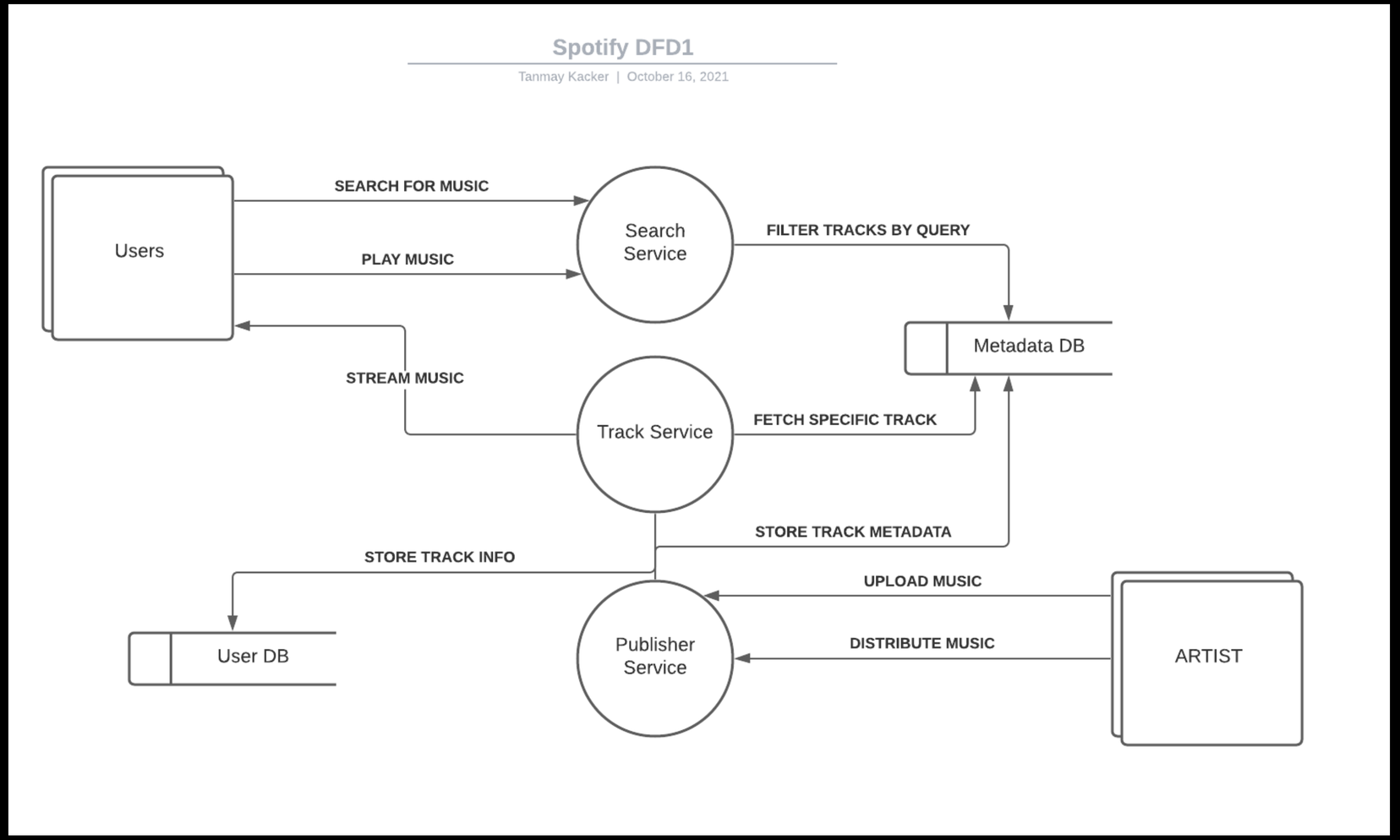
Spotify

Data Models



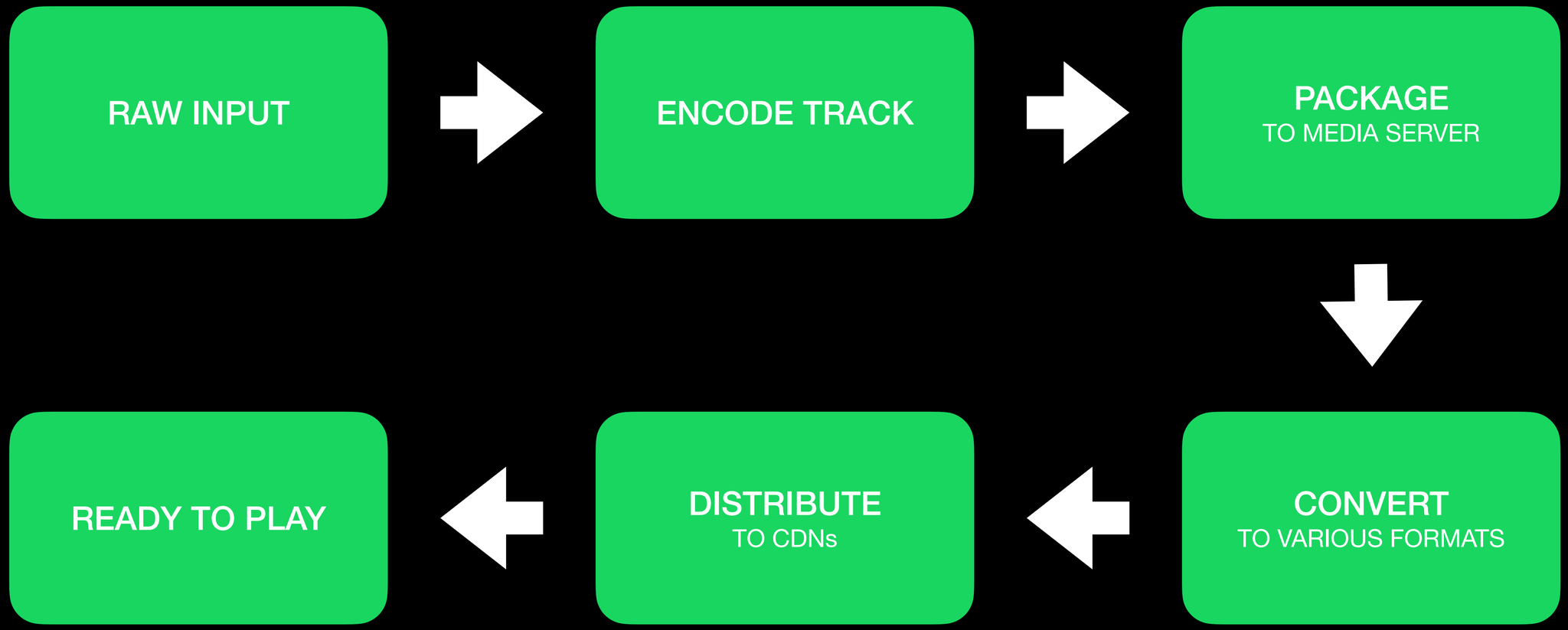
Spotify

Data Flow (DFD1)



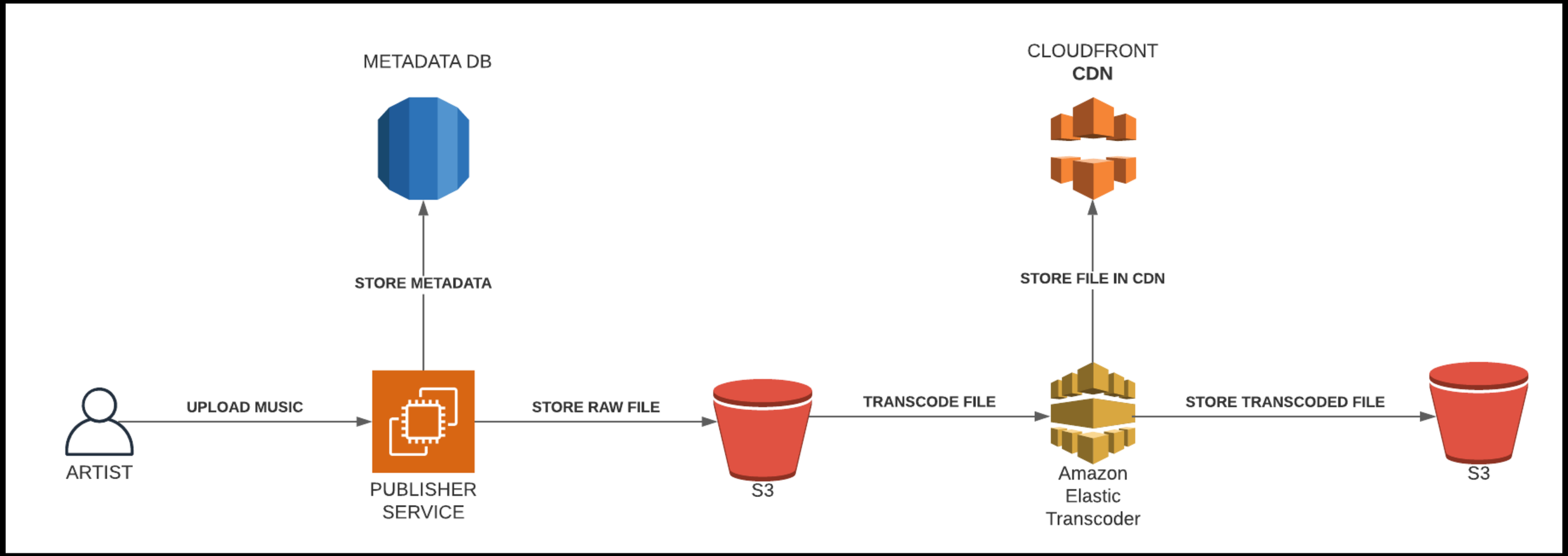
Spotify

Artist to Listener



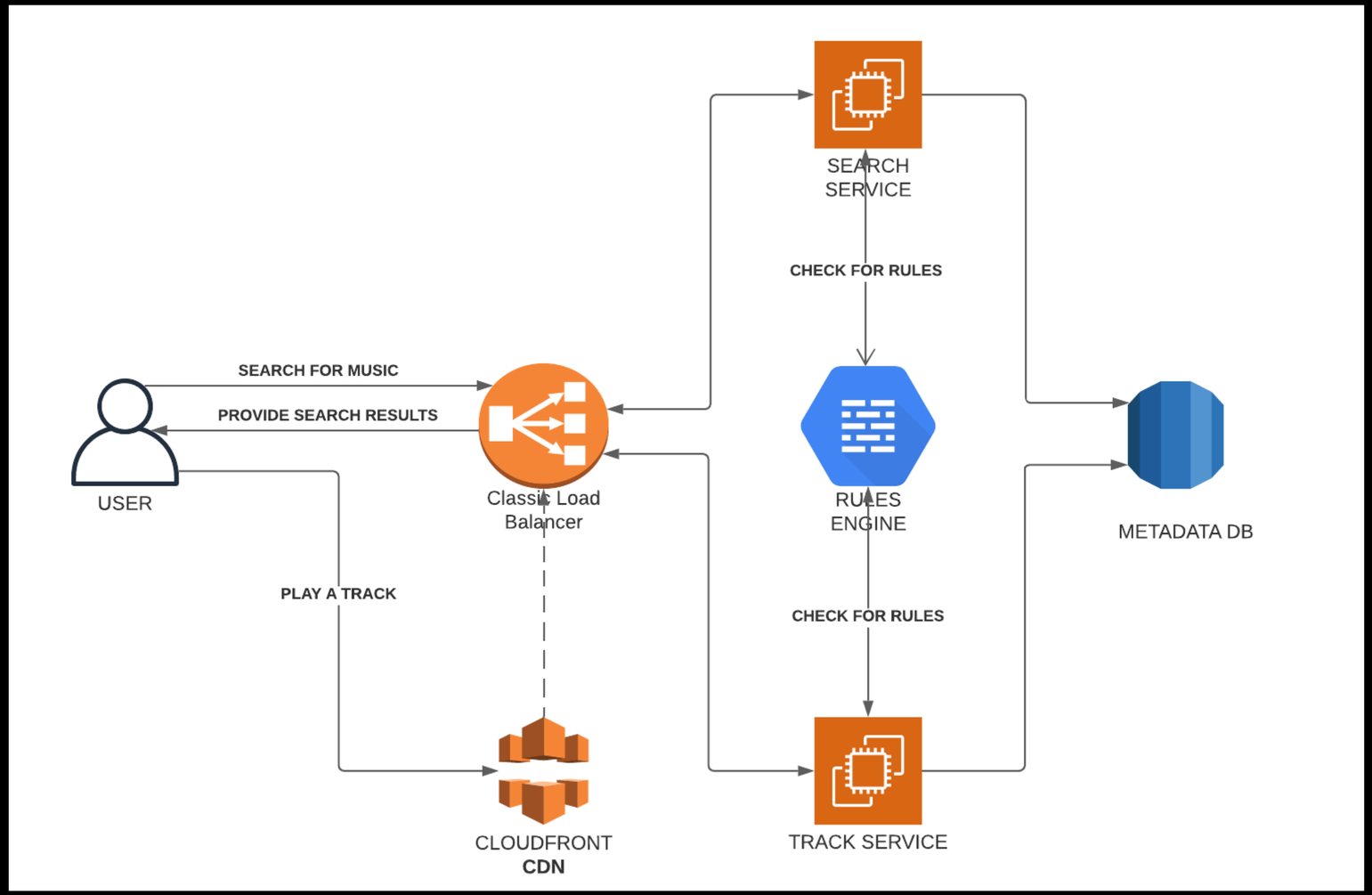
Spotify

Artist to Spotify



Spotify

Server to Listener



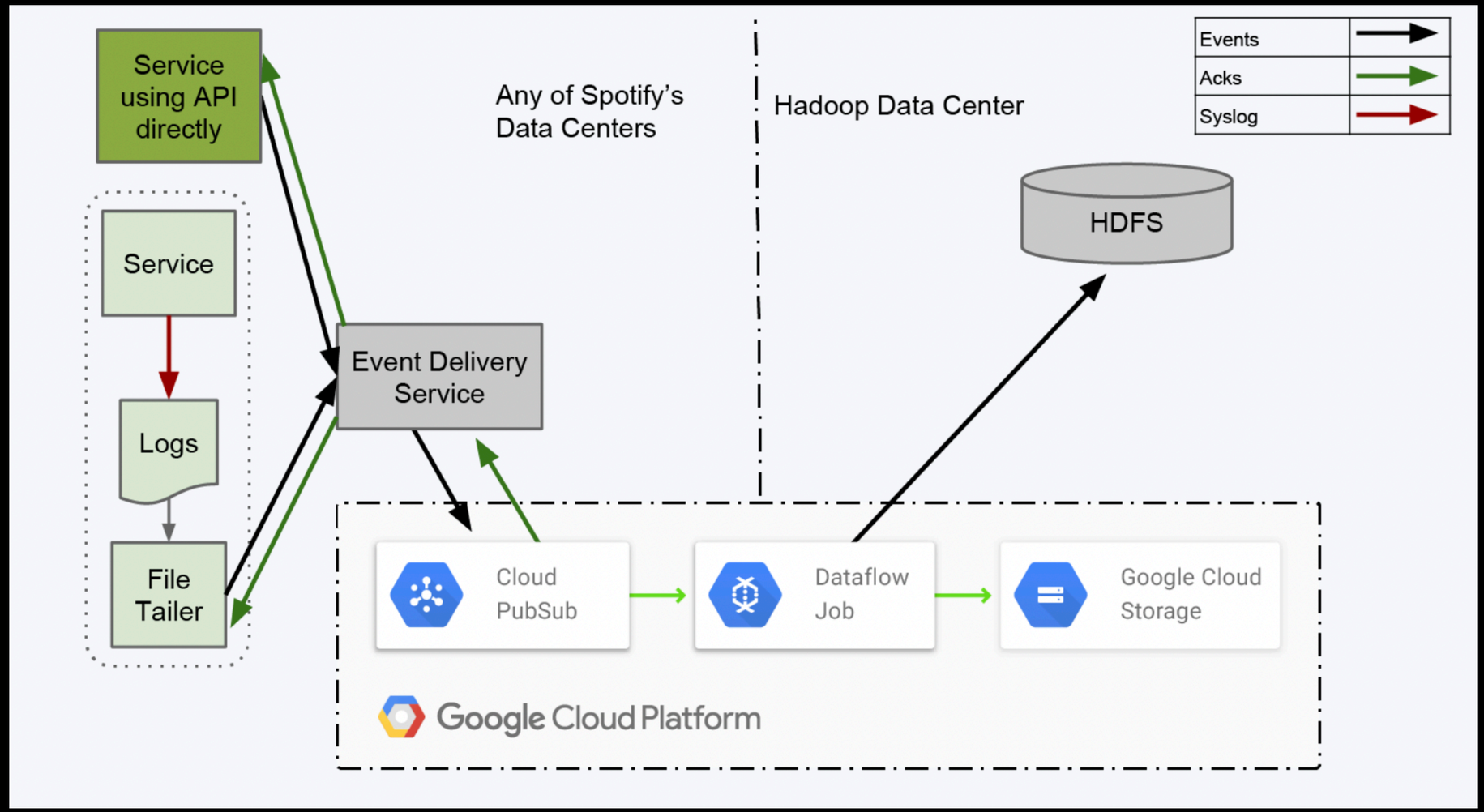
Spotify

X-factor

- Spotify stores music on various servers
- To stream a random track on your mobile, Spotify finds it on one of the millions of servers and sends it to your device
- Servers, in this case, are computers and all other gadgets of Spotify's subscribers. P2P
- Reduces playback latency

Spotify

Bonus - Event delivery system



Further reading

Spotify

- Rules engine
- Spotify's peer to peer
- Streaming protocols

References

Spotify

- System Design: Spotify
- Spotify's Event Delivery - I
- Spotify's Event Delivery - II

Thank You!

tanmaykacker40@gmail.com