Divide and Conquer:
On towards Microservices

Who am I?

My name is Gaurav

I work on:

- Philips Wellcentive Data Management Platform
- Docker, Jenkins, and Azure DevOps

What drove me to learn about microservices?

- Rise of technologies that are good at small tasks, namely Python, Node.js, etc.
- Cloud Native computing is the soul of most internet as we see it today.
- What looks a big chunk is actually a couple of finely granulated services working together.

Session outline

- Expectation from the session.
- Evolution of Software.
- WHATs and WHENs of Microservices.
- Who all have adapted to Microservices?
- HOWs of Microservices.
- What are the challenges?
- Q & A
- Demos and Examples.

What to expect from this session?

At the end of this session, the attendees will be able to:

- Understand what microservices are and how they make life better (Or tougher?)
- Understand various ways to achieve microservices.
- Understand the trend of granularity in software.
- Identify the requirements and evaluate the outcomes of microservices.

Software.. how it has been.

- Software has become more intricate compared to how it was earlier. The numbers of users have increased and with the gigantic code base, the complexity was undeniable. Other reasons were slower delivery, dependencies, tight coupling, etc.
- In early 2000s, Service-oriented architecture (SOA) started coming into picture, which slowly transitioned into hundreds of small independent services
- These services are what we know today as microservices.

Um, okay.. but what’s SOA?

- Service-oriented architecture (SOA) is a style of software design where services are provided to the other components by application components, through a communication protocol over a network.
- These are the 4 characteristics of a service in SOA:
    - It logically represents a business activity with a specified outcome.
    - It is self-contained.
    - It is a black box for its consumers.
    - It may consist of other underlying services.

Source: Wikipedia

That's great, but what do I get from using SOA?

- **Business value** is given more importance than technical strategy.
- **Strategic goals** are given more importance than project-specific benefits.
- **Intrinsic inter-operability** is given more importance than custom integration.
- **Shared services** are given more importance than specific-purpose implementations.
- **Flexibility** is given more importance than optimization.
- **Evolutionary refinement** is given more importance than pursuit of initial perfection.

Source: Wikipedia

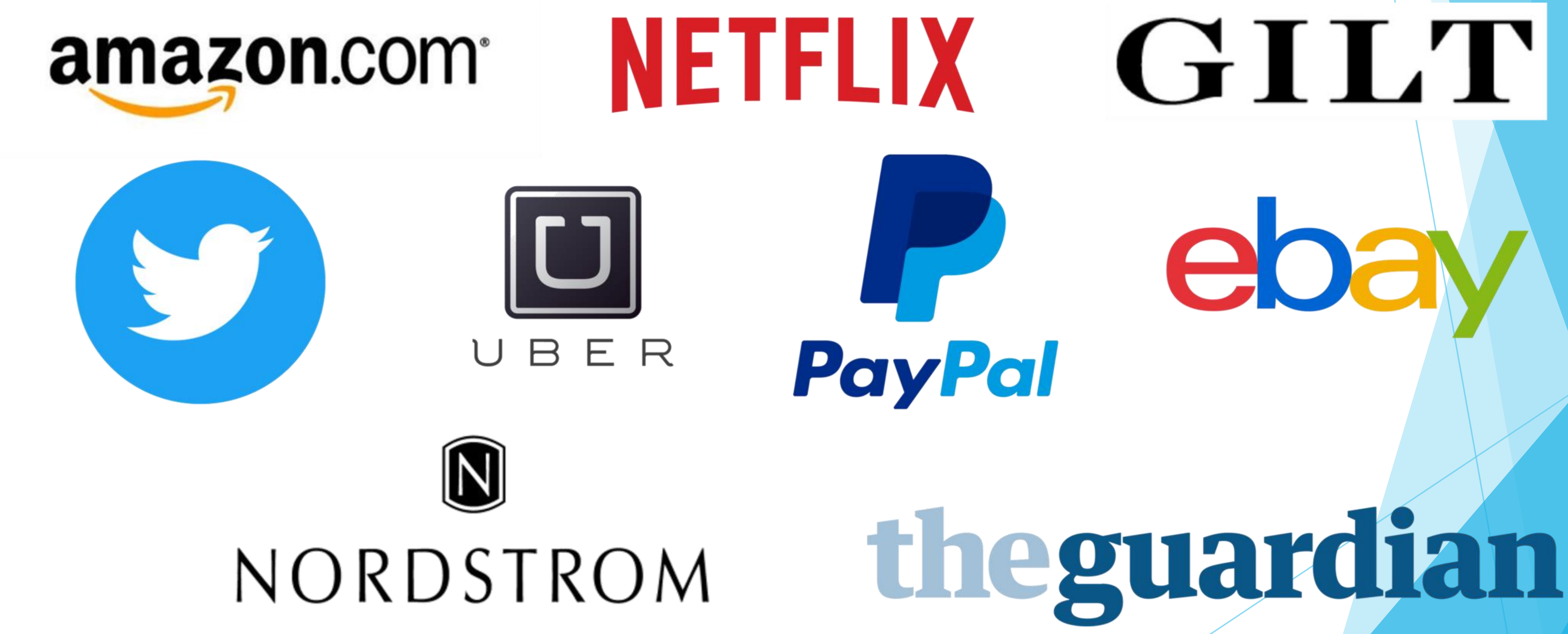
But we were here for a session on microservices, what is this?

- Fine grained version of SOA
- Loosely coupled and independent of each other
- They are small (potentially and very often a process)
- Several independent microservices work together to perform a bigger task
- Independently deployable
- Enables software to be horizontally scalable with ease
- Helps in decentralizing continuous delivery
- Language independent

Oh, where did the they come from?

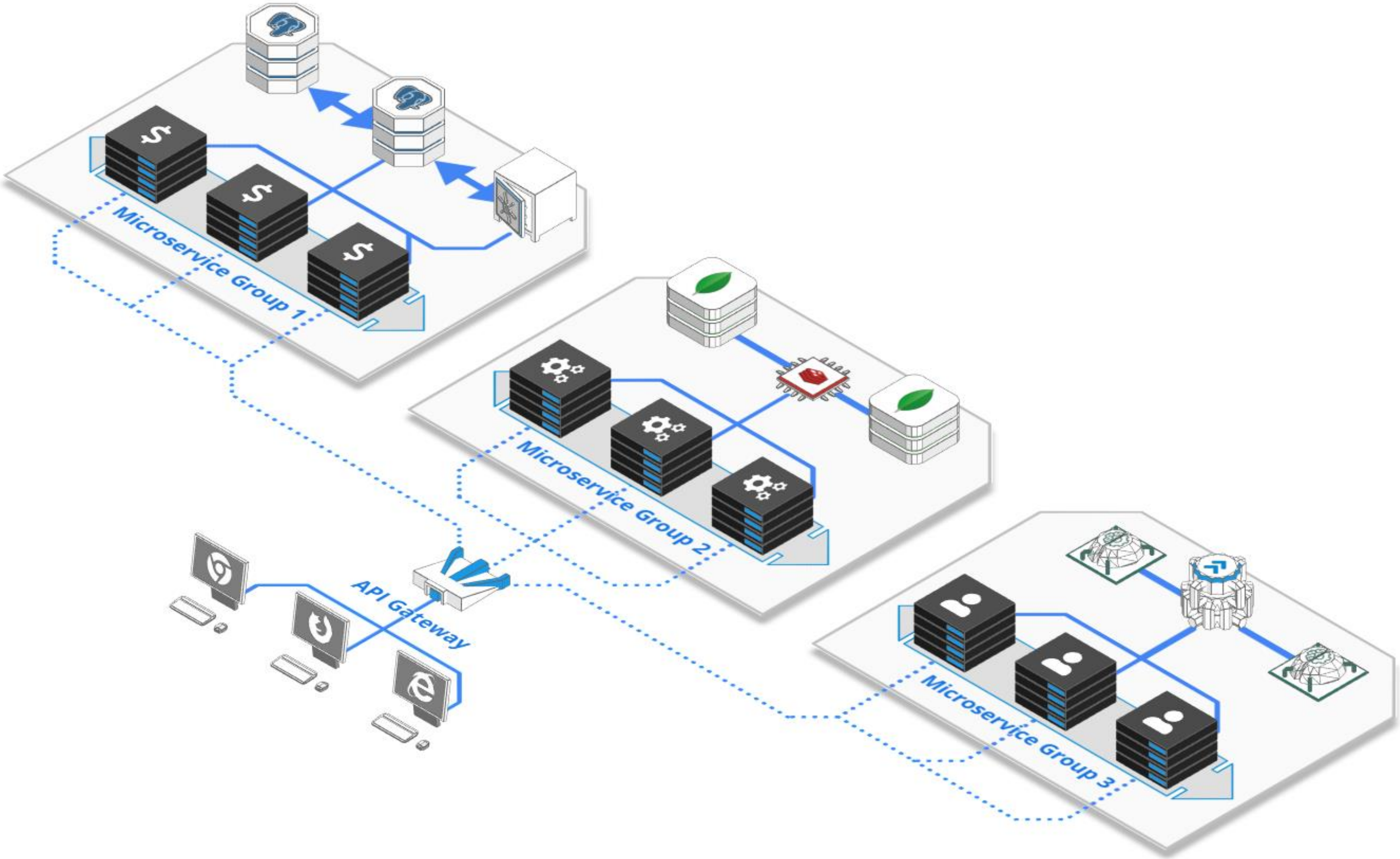
- “Microservices”, the term, was coined in 2011 in a workshop of Software Architects in Venice.
- Microservices are inspired by Unix’s idea of loose-coupling and way of working, “Do one thing and do it well”.

Who all have adapted to Microservices?



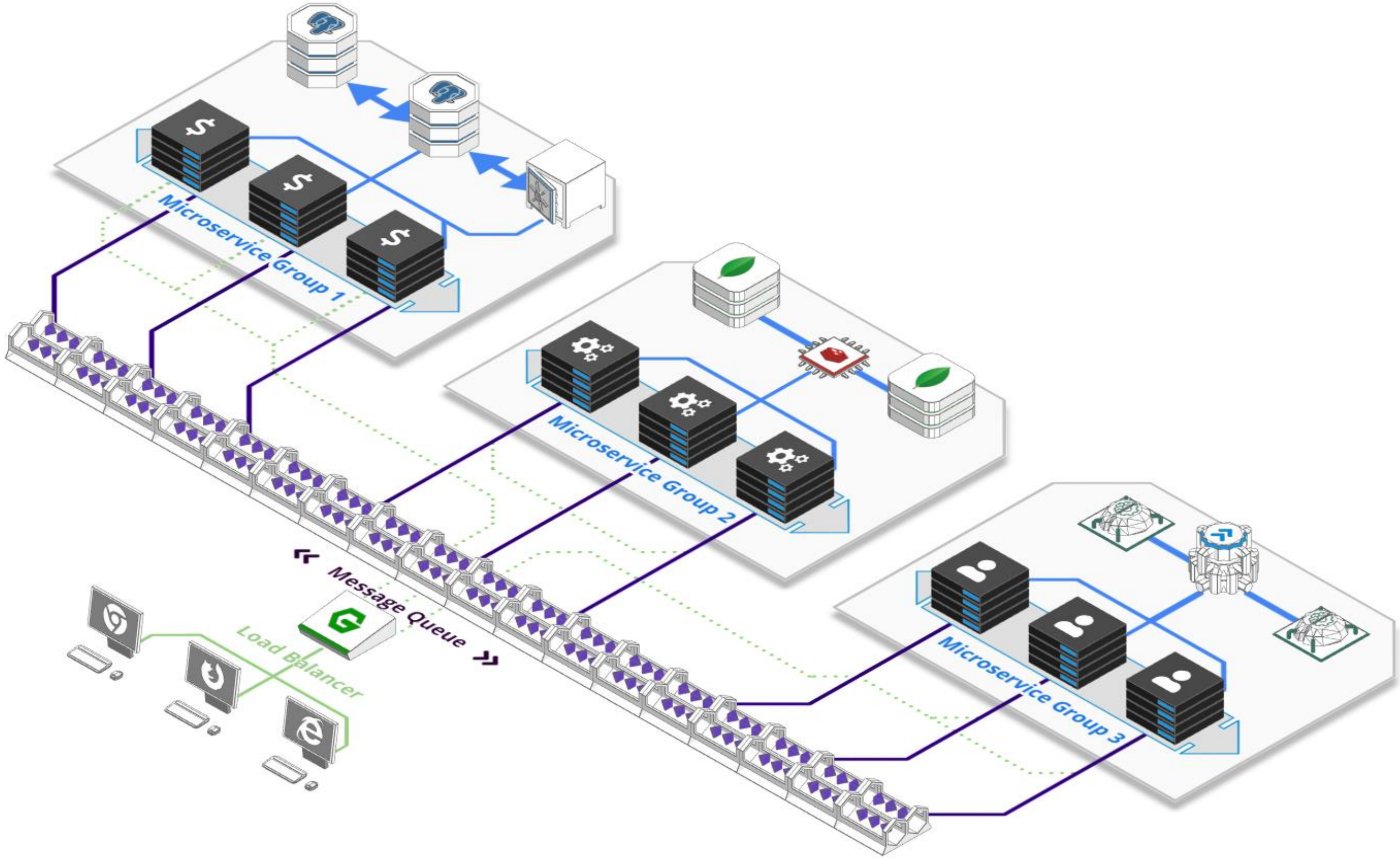
And many more…

They can be achieved in various ways – API Gateway



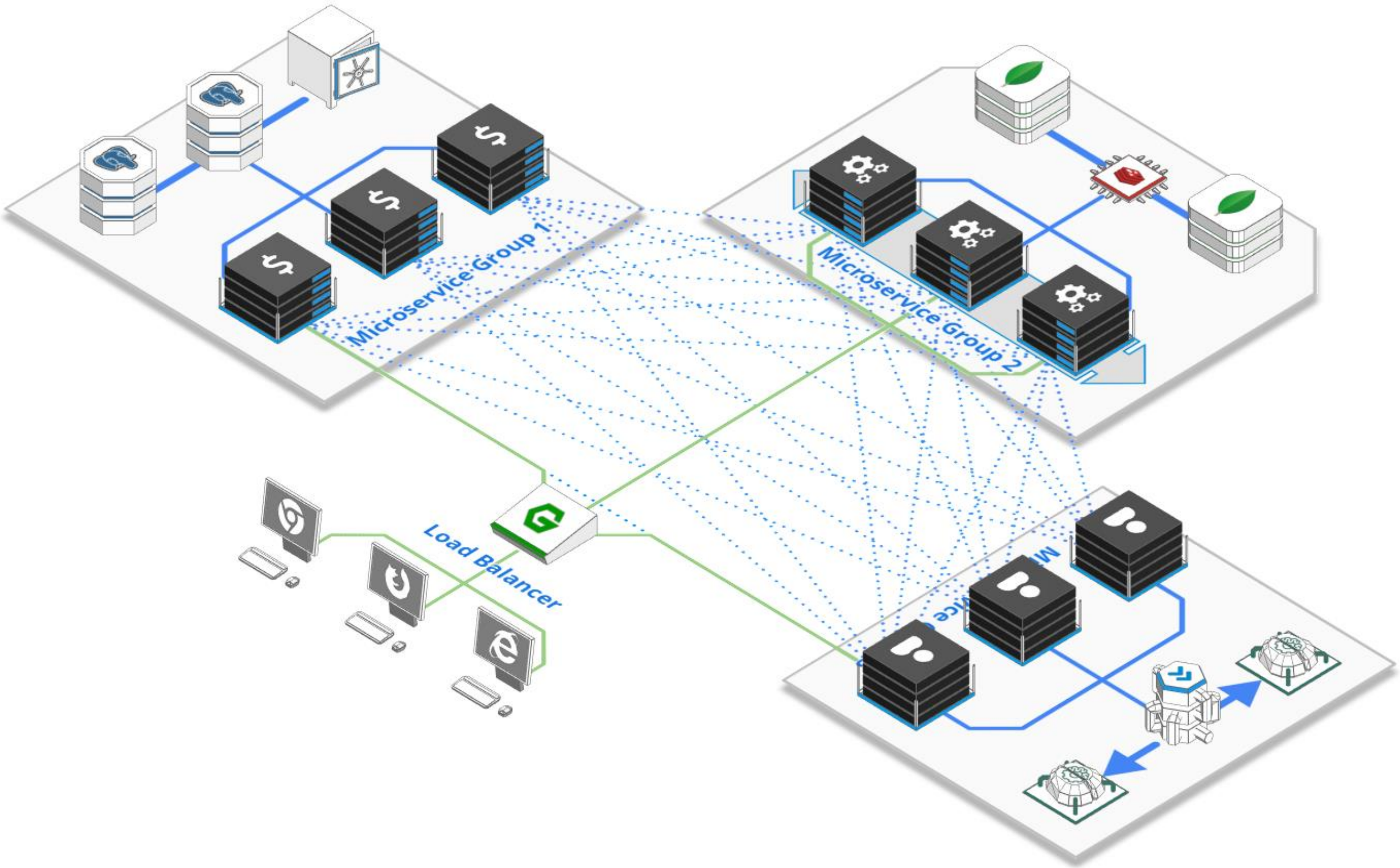
Source: Arcentry.com

They can be achieved in various ways – Message Queues



Source: Arcentry.com

They can be achieved in various ways – Service mesh

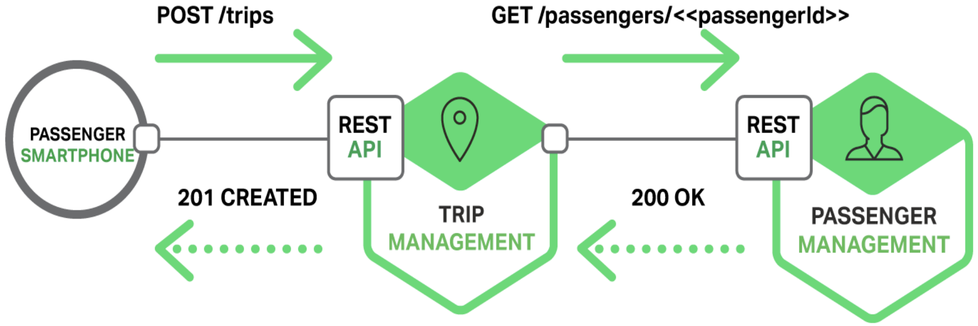


Source: Arcentry.com

How do the services talk?

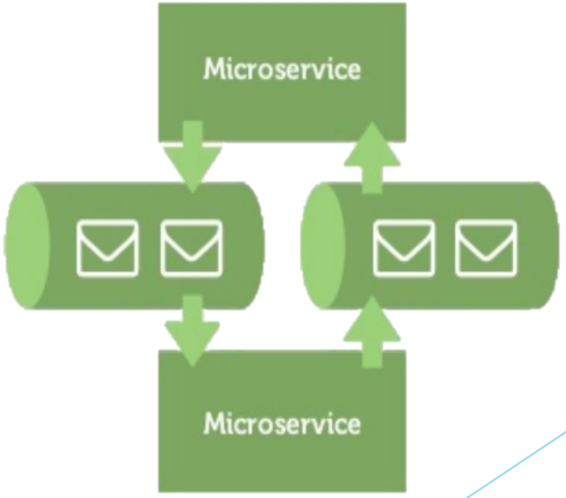
Services can talk synchronously as well as asynchronously.

- Synchronous:
  e.g. REST based



- Asynchronous:

e.g. Messaging based



This can’t be perfect! Tell me the downs now!

- Developing distributed systems can be complex. Since everything is now an independent service, you have to carefully handle requests traveling between your modules.
- Transaction management can be painful.
- Testing a microservices-based application can be cumbersome.
- Deploying microservices can be complex.

I’ve got this! What are my weapons?

- Any general-purpose programming language and related data persistence strategy.
- A communication medium for your services.

Q & A