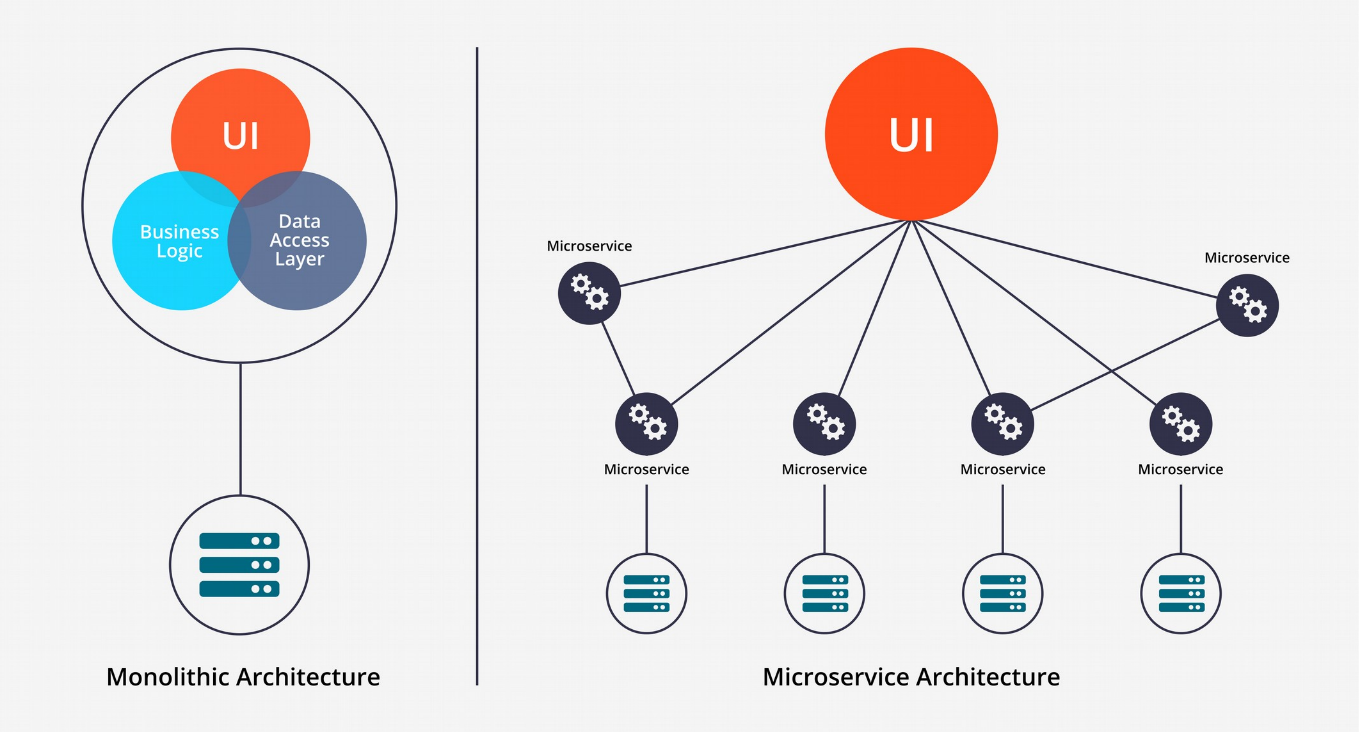
Enterprise Architecture

Design

Remember Microservices ?

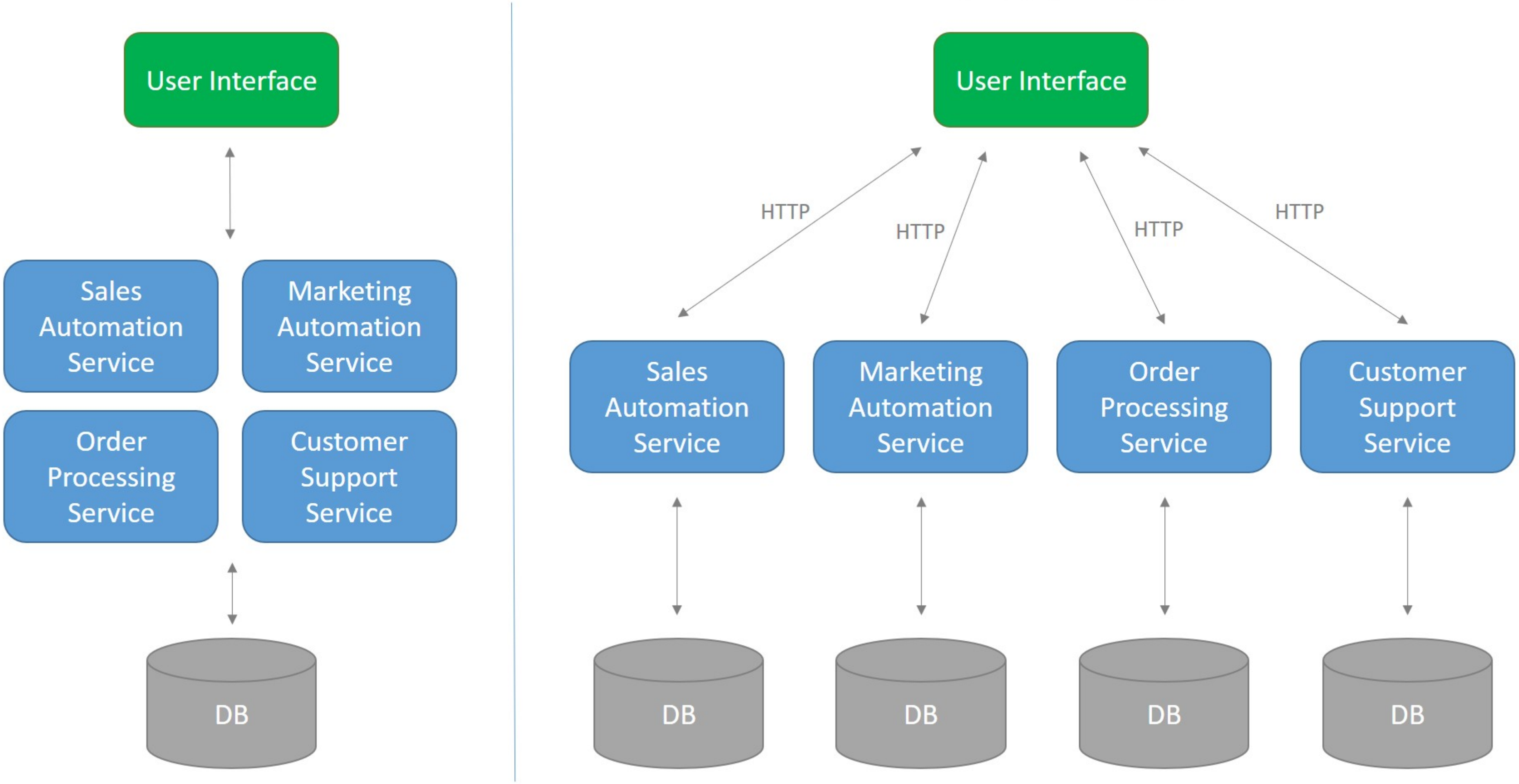
- Deployed Independently, has its own process(es)
- Small team to create and manage
- Does one thing really well, communicates with others when it needs to



https://www.weave.works/blog/what-are-microservices/

Monolith

Microservices



User Interface

Sales Automation Service

Marketing Automation Service

Order Processing Service

Customer Support Service

DB

User Interface

HTTP

HTTP

HTTP

HTTP

Sales Automation Service

Marketing Automation Service

Order Processing Service

Customer Support Service

DB

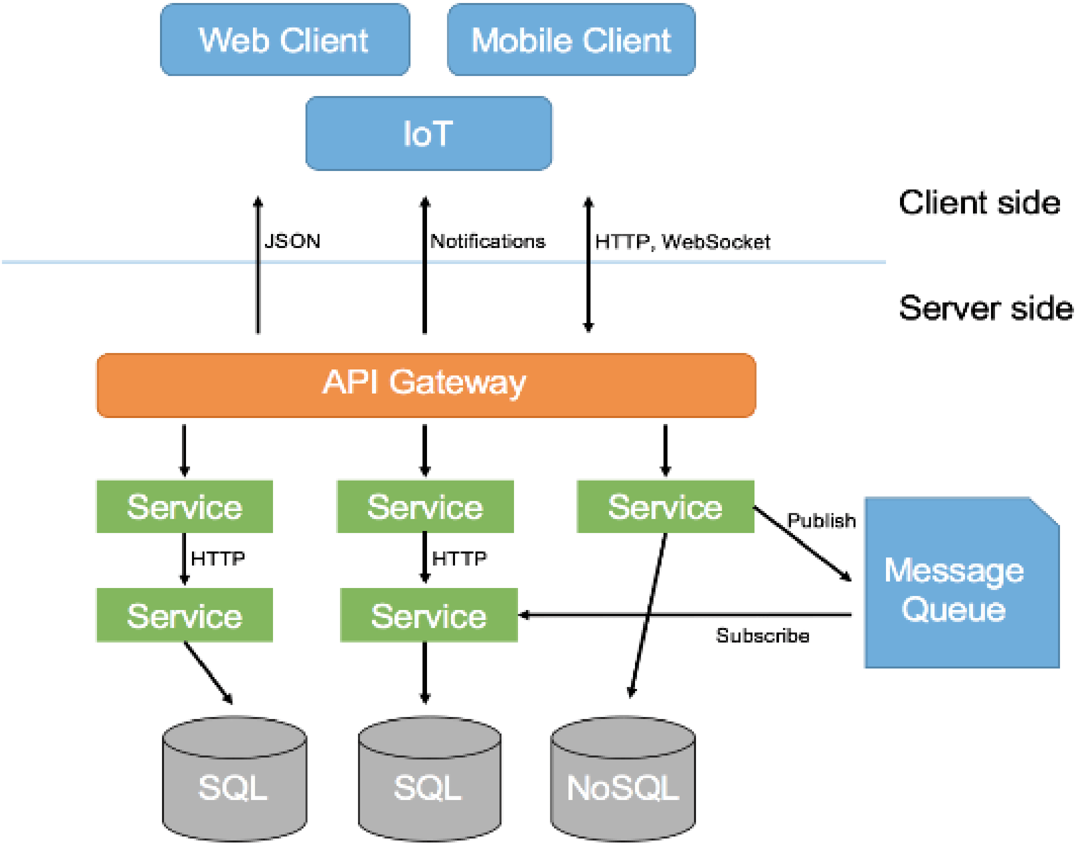
DB

DB

DB

https://www.carlrippon.com/scalable-and-performant-asp-net-core-web-apis-microservices/

API Gateway



https://www.techjini.com/blog/microservices/

Do you notice any communication patterns ?

Two Architectural decisions → Design

- How do we determine the scope of each microservice ?!
- How should they communicate ?!
    - Direct API calls ?
    - Events on a shared queue ?
    - Something else ...?

Q1) How do we design APIs ?

API vs Microservice

- Watch:
  - https://www.youtube.com/watch?v=WFtmCaqIjkA

Designing API

What are the main architectural styles for APIs?

API Styles

- Rest
  - HATEOAS
- RPC
  - gRPC
  - SAOP
- GraphQL
  - Falcor

REST

- Representational state transfer
- Very common on web over HTTP methods
  - GET,POST,PUT,..
- Generally stateless calls

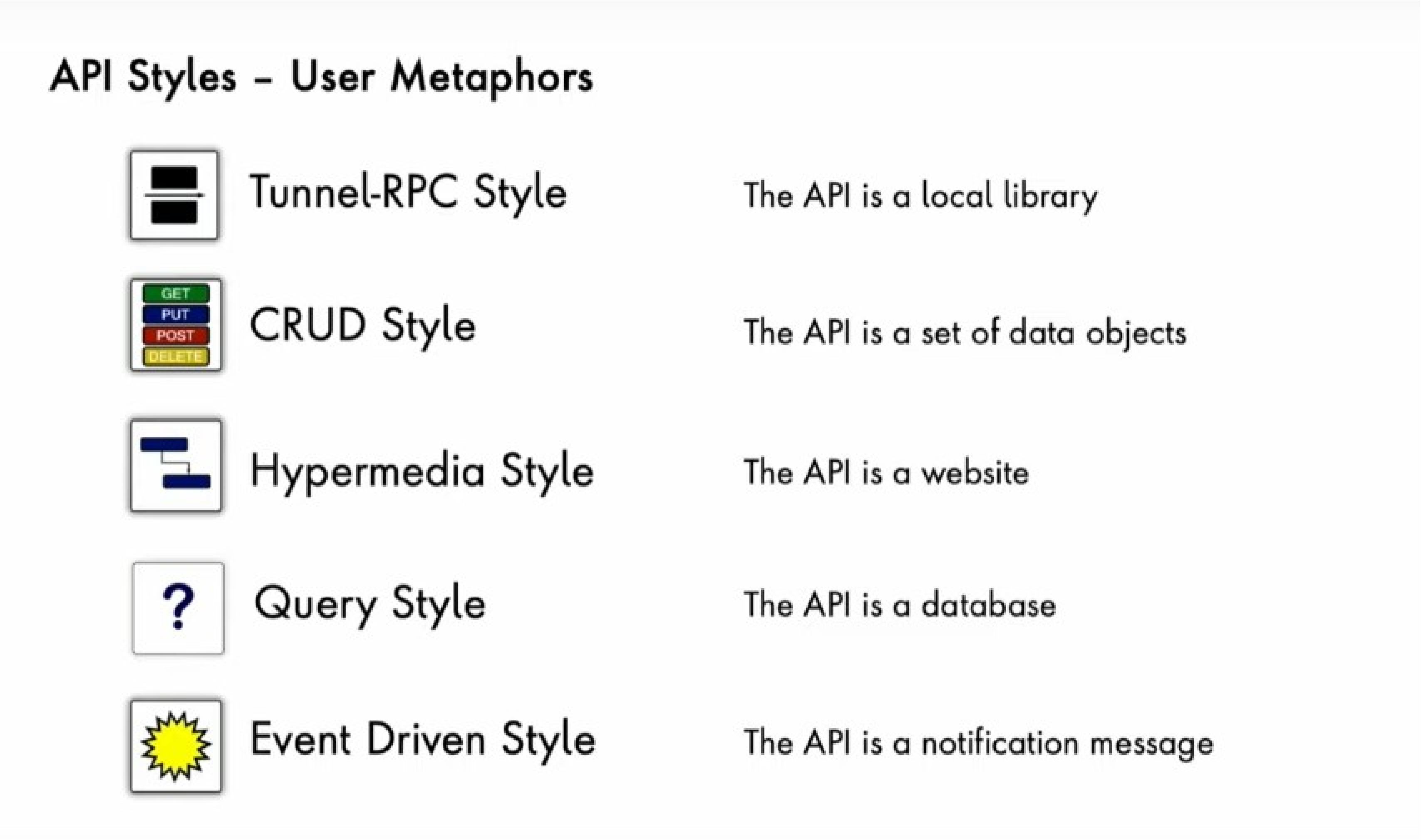
gRPC

- Remote Procedure calls, are efficient and strongly typed
- gRPC is optimized:
  - Transport over http/2
  - Binary protocol & serializing (e.g ProtoBuf)
- Messages support schema evolution
- Bi-directional communication/streaming
- Checkout:
  - https://www.grpc.io/

GraphQL

- Query parts of JSON like object
- Clients get exactly what they ask for and nothing more!
- Checkout:
  - https://graphql.org/

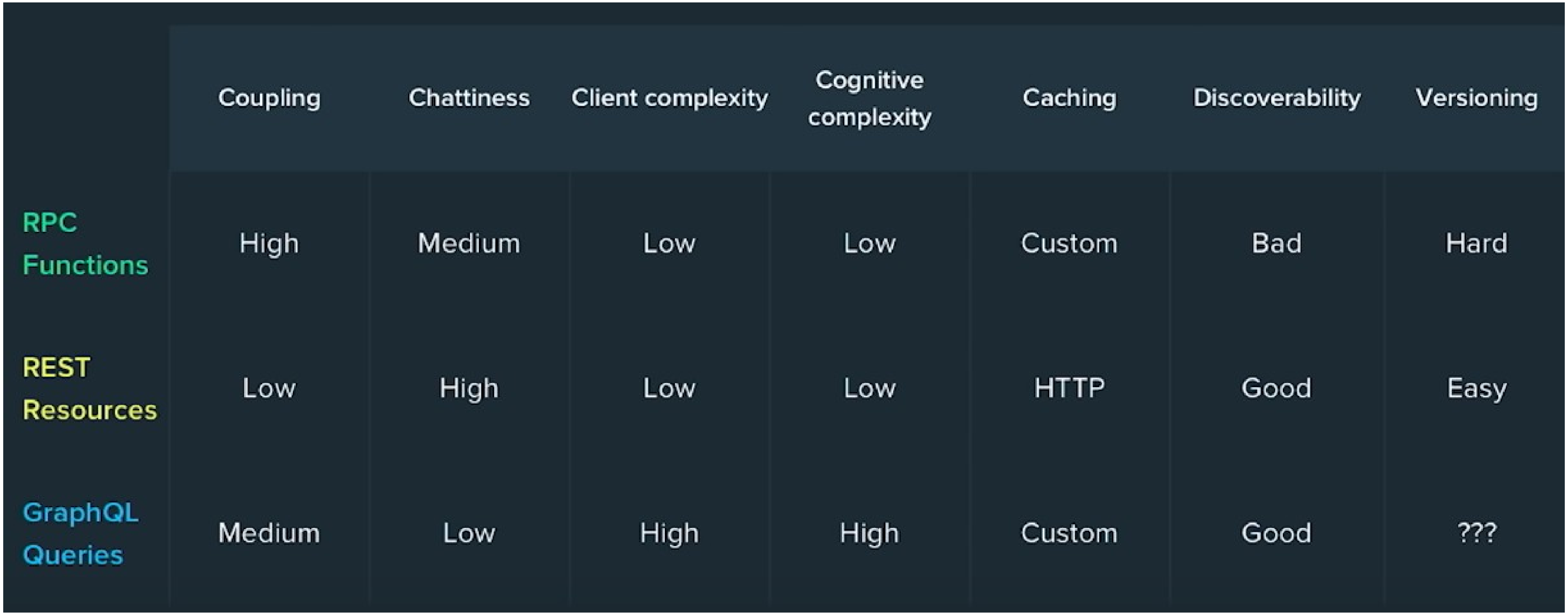
API Styles



API Styles – User Metaphors

Tunnel-RPC Style	The API is a local library
CRUD Style	The API is a set of data objects
Hypermedia Style	The API is a website
Query Style	The API is a database
Event Driven Style	The API is a notification message

Trade offs



	Coupling	Chattiness	Client complexity	Cognitive complexity	Caching	Discoverability	Versioning
RPC Functions	High	Medium	Low	Low	Custom	Bad	Hard
REST Resources	Low	High	Low	Low	HTTP	Good	Easy
GraphQL Queries	Medium	Low	High	High	Custom	Good	???

Nate Barbettini – API Throwdown: RPC vs REST vs GraphQL, Iterate 2018

API styles

- Let’s watch (45 min talk)
  - Nate Barbettini – API Throwdown: RPC vs REST vs GraphQL, Iterate 2018
    - https://youtu.be/IvsANO0qZEg?t=75

Designing API

- Read:
  - https://api-university.com/blog/styles-for-apis-soap-rest-and-rpc/

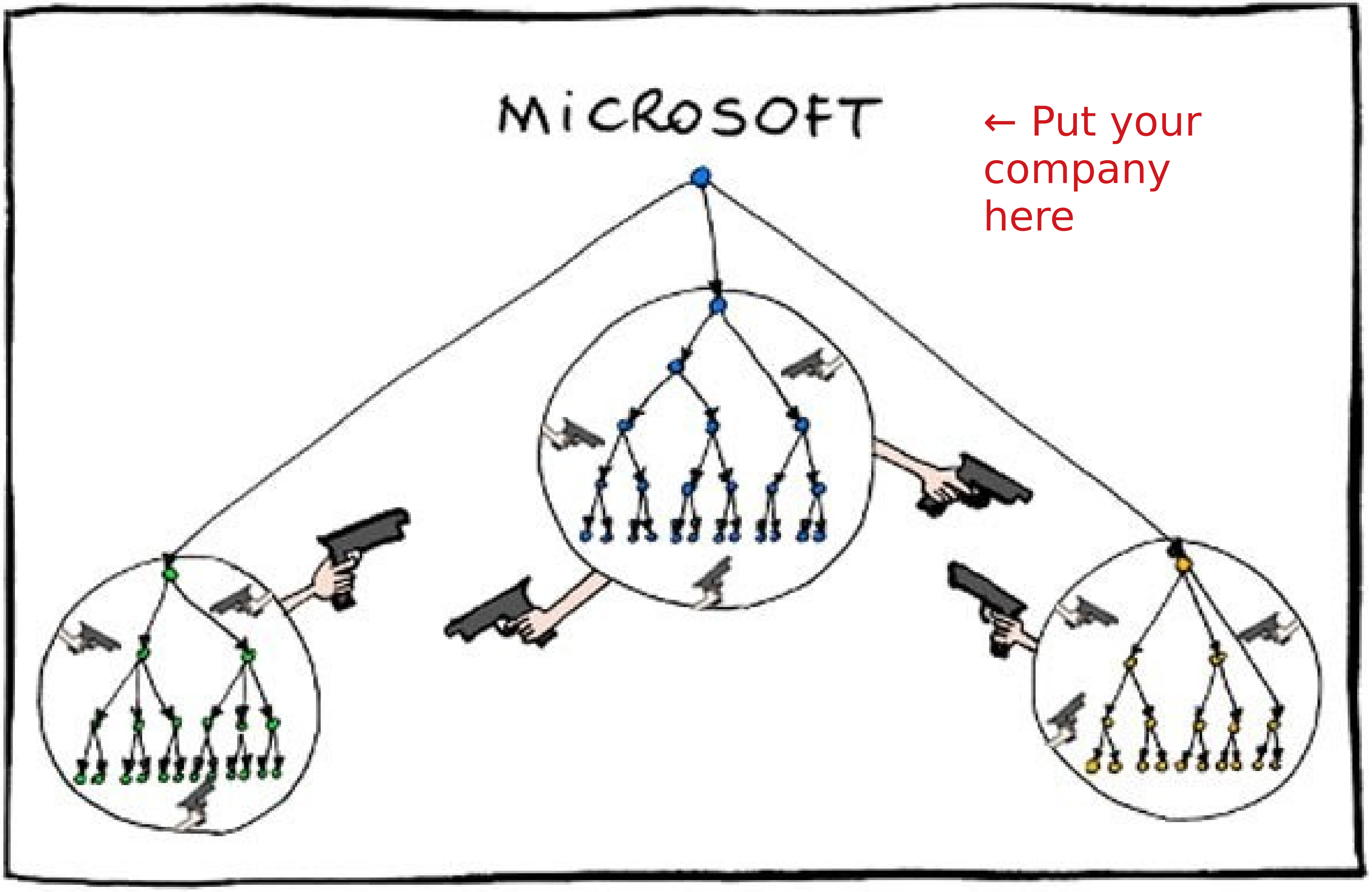
Designing API

- Got time?, Watch:
  - Designing Quality APIs (Cloud Next '18)
    - https://www.youtube.com/watch?v=P0a7PwRNLVU
  - GOTO 2019 • Practical API Design • Ronnie Mitra
    - https://www.youtube.com/watch?v=272ZZ53HS_4

Q2) How do we divide our monolith into microservices ?

Conway’s Law

Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations!



← Put your company here

http://www.businessinsider.com/the-ballmer-re-org-chart-2013-7