

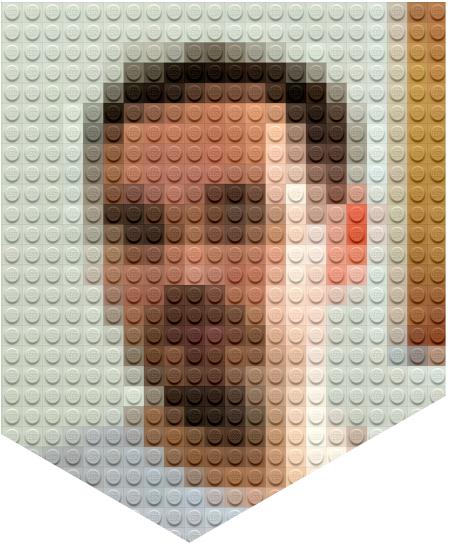


Et si les micro-services n'avaient
rien à voir avec la technique ?

AlpesCraft 2022

Yves Brissaud

@_crev_



Yves Brissaud

Senior Software Engineer @ Docker

- Docker Hub
- Vulnerability Scanning
- Publisher experience:
 - Docker Official Images
 - Docker Verified Publishers
 - Docker sponsored Open Source Program



- Délivrer de la valeur
- En continu
- Durablement
- Rapidement



Microservices vs Monoliths vs ...



“Start with monolith and extract micro services”

— Tammer Saleh

“Don’t start with a monolith when your goal is a micro services architecture”

— Stefan Tilkov

“If you can’t build a monolith, what makes you think micro services are the answer?”

— Simon Brown

Micro service: something that could be rewritten in two weeks

— Jon Eaves



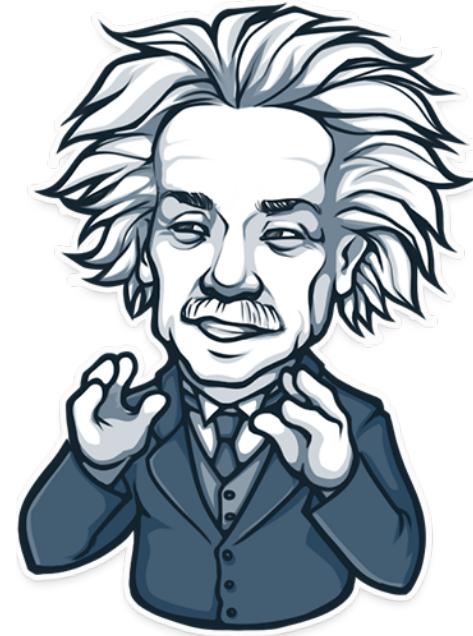
Ce que vous ne trouverez pas ici

- ~~Outil, framework, ...~~
- ~~Architecture~~
- ~~Caractéristique, taille, ...~~
- ~~Docker, container, kubernetes, ...~~



Et si les micro-services avaient parfois à voir avec la technique ?

- Besoin en ressources spécifique
- Besoin en sécurité spécifique
- Besoin technique spécifique
- Métier spécifique
- ...



Si les micro-services n'ont (généralement) rien à voir avec la technique, avec quoi alors ?



Si les micro-services n'ont (généralement) rien à voir avec la technique, avec quoi alors ?

Les humains



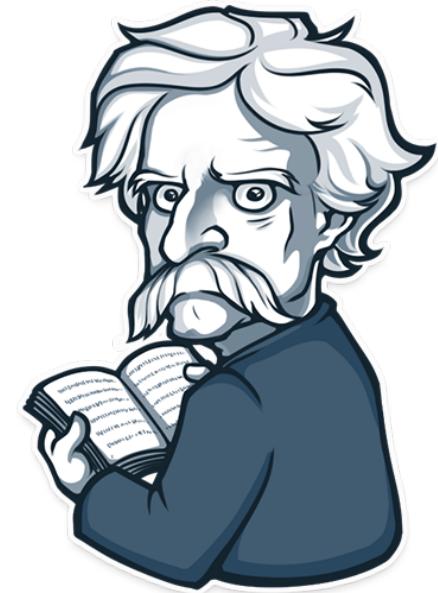
La loi de Conway



Définition

« Les organisations qui conçoivent des systèmes,
tendent *inévitablement* à produire des designs
qui sont des copies de la *structure de communication* de leur organisation. »

— Melvin Conway, 1968



Pourquoi cette loi compte ?

« Des différences significatives dans la modularité du produit sont cohérentes avec le fait que des équipes distribuées ont tendance à développer des produits plus modulaires »

— MIT - Harvard Business School “Exploring the Duality between Product and Organisational Architectures”

« Si l'architecture du système et l'architecture de l'organisation sont en désaccord, l'architecture de l'organisation gagne »

— Ruth Malan

On a toujours le choix

- Contraindre le design technique
-> assumer les conséquences humaines
- Contraindre l'organisation humaine
-> assumer les conséquences techniques



La manoeuvre inverse de Conway

Organiser les équipes pour atteindre le design
technique souhaité



Organiser les équipes requiert une expertise technique

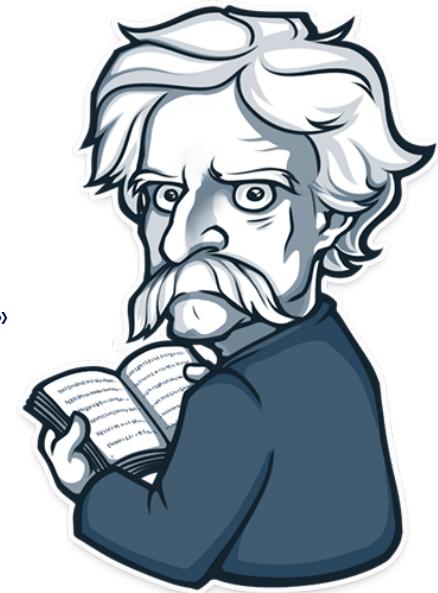
Si nous avons des managers qui décident quels services vont être créés, par quelle équipe, nous avons implicitement des managers qui décident de l'infrastructure système

— Ruth Malan

Définition

« Les organisations qui conçoivent des systèmes,
tendent *inévitablement* à produire des designs
qui sont des copies de la **structure de communication** de leur organisation. »

— Melvin Conway, 1968



Communication entre services

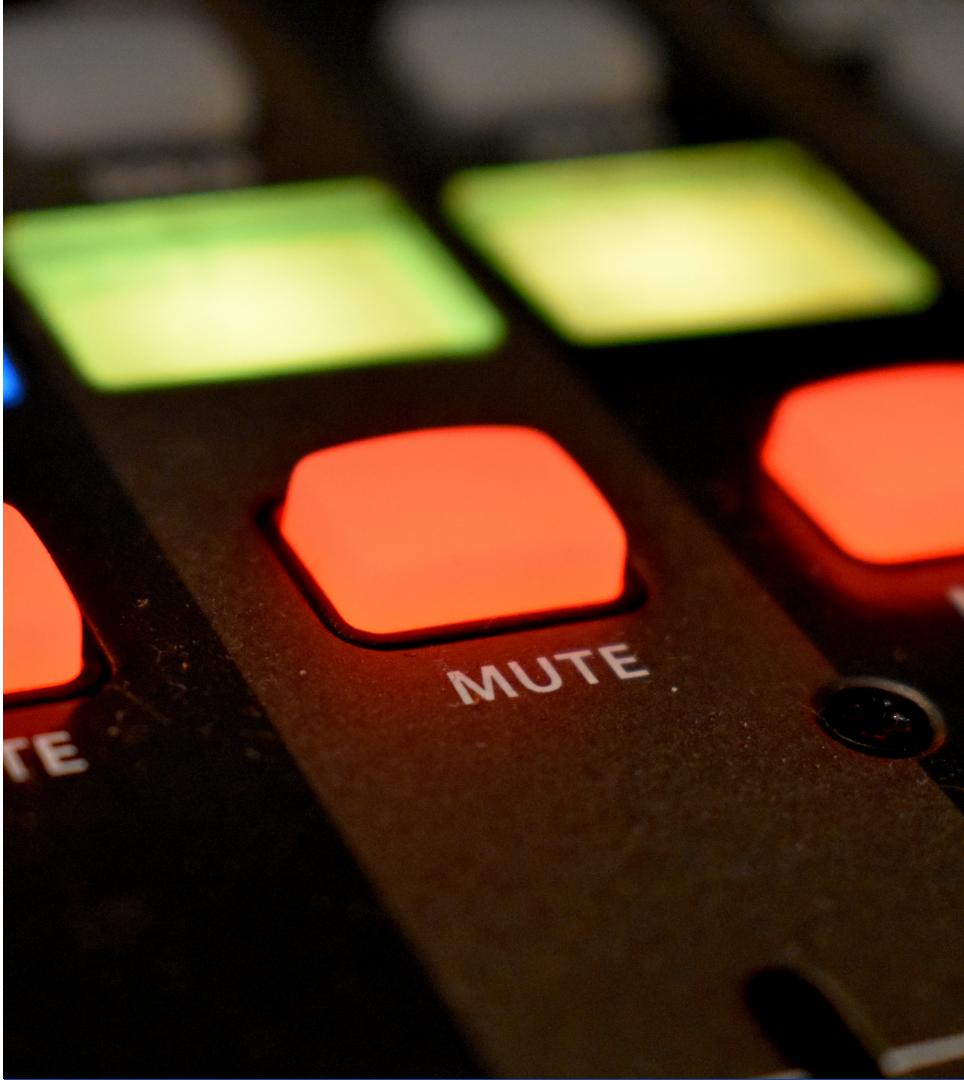
- Qui communique avec qui
 - Type de communication (API, event bus, ...)
 - Format
 - ...
- ➡ définir, restreindre, contractualiser la communication



Et pour les équipes ?

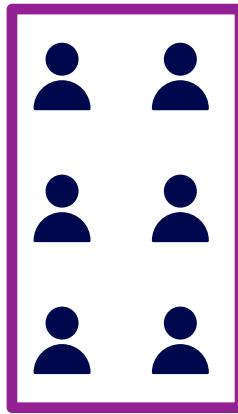


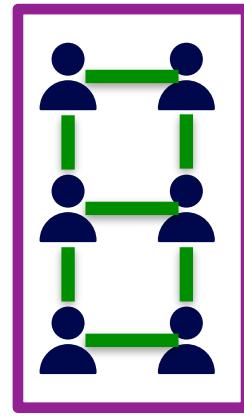
Définir, restreindre, contractualiser
la communication



La communication ?

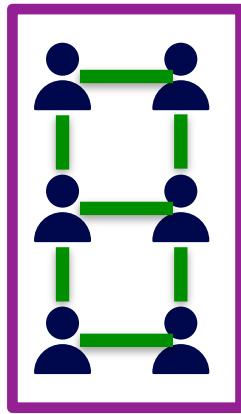
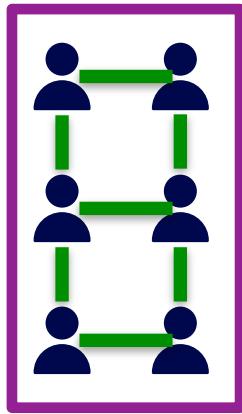






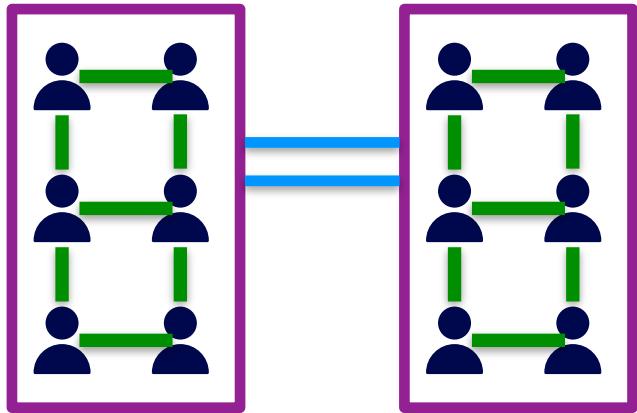
High bandwidth





High bandwidth



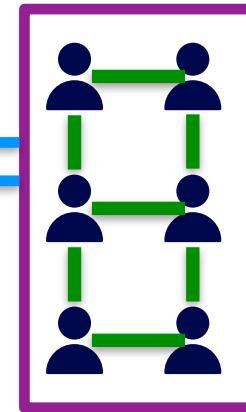
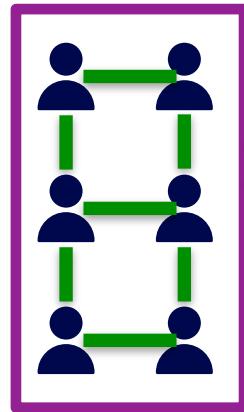
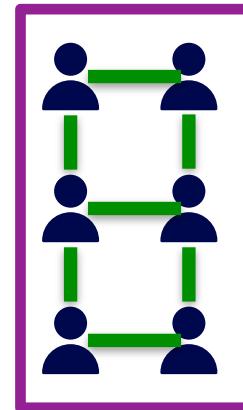
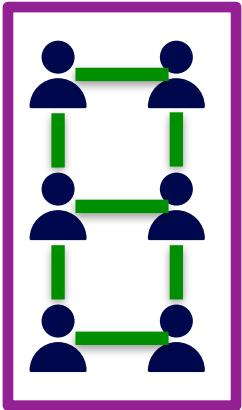


High bandwidth



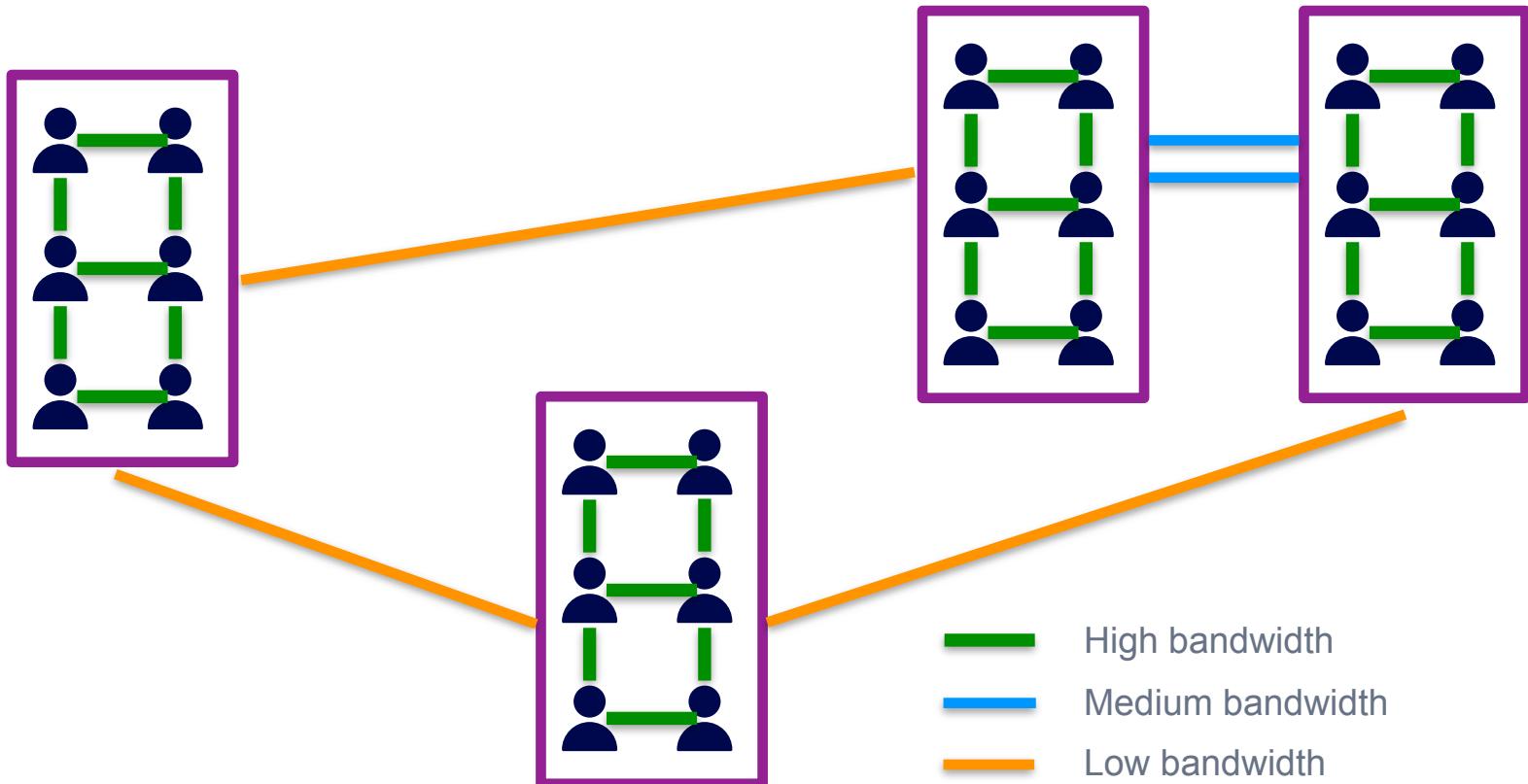
Medium bandwidth





High bandwidth
 Medium bandwidth





OK, en fait tu nous
réinventes les silos, non ?



OK, en fait tu nous
Non
réinventes les silos, non ?



- Interactions ?
- Dépendances ?

Entre les équipes

- Comment ?
- Pourquoi ?

Communiquer avec une équipe



Formaliser pour ne pas subir



Team API

- Qui sommes nous ?
- Que faisons nous ?
- Que produisons nous ?
- Comment travaillons nous ?
- Comment nous contacter ?
- Avec quelles équipes interagissons-nous ?
- ...

<https://github.com/TeamTopologies/Team-API-template>

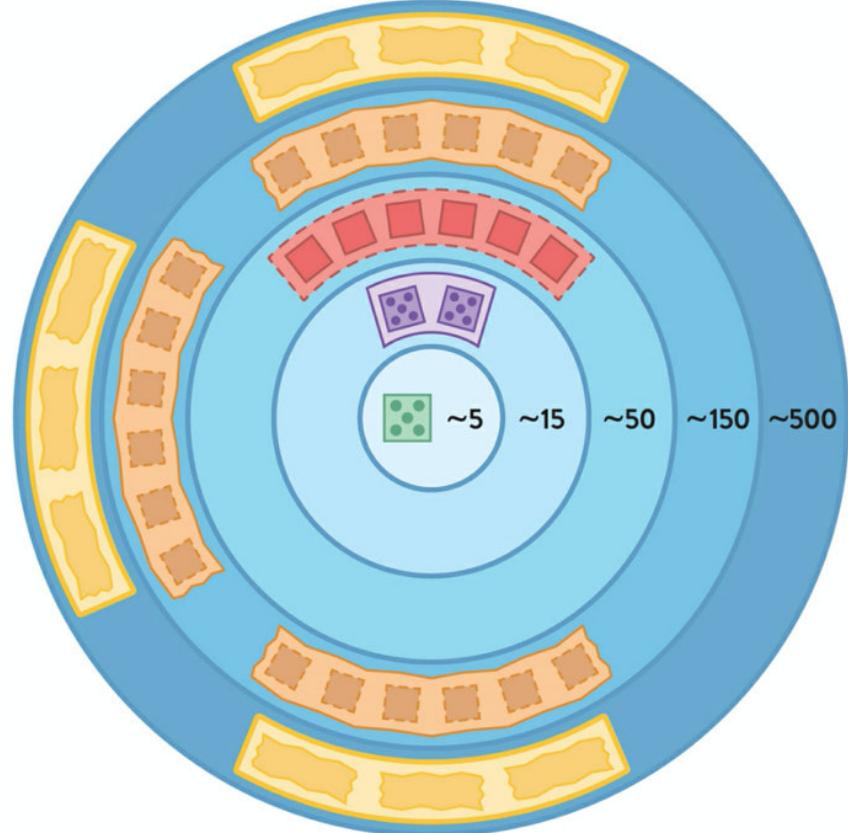


C'est comment une équipe ?



Taille

- Nombre de relations
- 🎲 Dunbar's numbers
 - 40% du temps social avec 5 personnes
 - +20% avec +10 personnes



Charge cognitive

- Charge intrinsèque : le code, sa complexité, sa compréhension, ...
- Charge extrinsèque : son déploiement, sa configuration, ...
- Charge essentielle : interactions avec les autres services, ...



Ownership

- L'équipe possède le logiciel (mais pas les individus)
- Responsabilité de tout le monde = responsabilité de personne
 - ➡ No shared ownership



Diversité

- ➡ Plus de créativité
- ➡ Variété des points de vues et expériences



Équipe et non individus

- Individus < dynamique de l'équipe

Re:Work – the five keys to a successful Google team

- État d'esprit équipe



Responsabilités

- Limiter les responsabilités à la charge cognitive que l'équipe puisse supporter







Team Topologies

Un “framework”,
pour l’organisation des équipes,
dans le but d’obtenir le design technique souhaité



4 topologies d'équipes, 3 modes de communication

Pour designer une organisation comme on
design un système



Les 4 topologies d'équipes



Stream-aligned team

Sur un segment métier

“you build it, you run it”

Principal



Enabling team

Aide, support aux stream aligned teams



Complicated subsystem team

Où une expertise spécifique est requise



Platform team

Fourni un produit interne

Accélère la livraison des autres équipes



Les 3 modes de communication





Collaboration

Sur une période donnée

Pour découvrir de nouvelles choses

Équipes travaillant de pair

2 équipes

Souvent au moins un stream aligned



Facilitation

Aide

Mentoring

Ressources supplémentaires

1 équipe (généralement enabling)
facilite une autre



X-as-a-service

Consomme le produit d'une autre :
API, outil, produit

Toutes les équipes sauf enabling



Réductions de la charge cognitive

- Meilleures pratiques, code :  charge intrinsèque
 - ➡ Enabling team
- Meilleurs outils, processus :  charge extrinsèque
 - ➡ Complicated subsystem, platform team
- Meilleures docs, team API :  charge essentielle
 - ➡ Expliciter X-as-a-service





A vous de jouer !

Définissez vos équipes
Explicitez les structures de communication



4 fundamental topologies

— Stream-aligned team

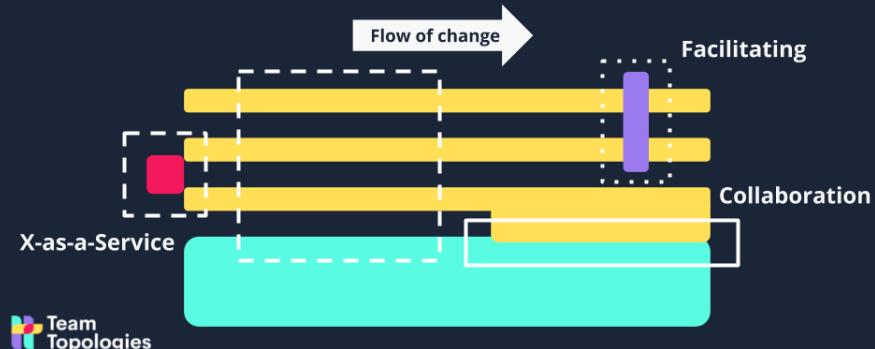
| Enabling team

■ Complicated Subsystem team

■ Platform team



3 core interaction modes



Et dans la pratique ?



Équipes

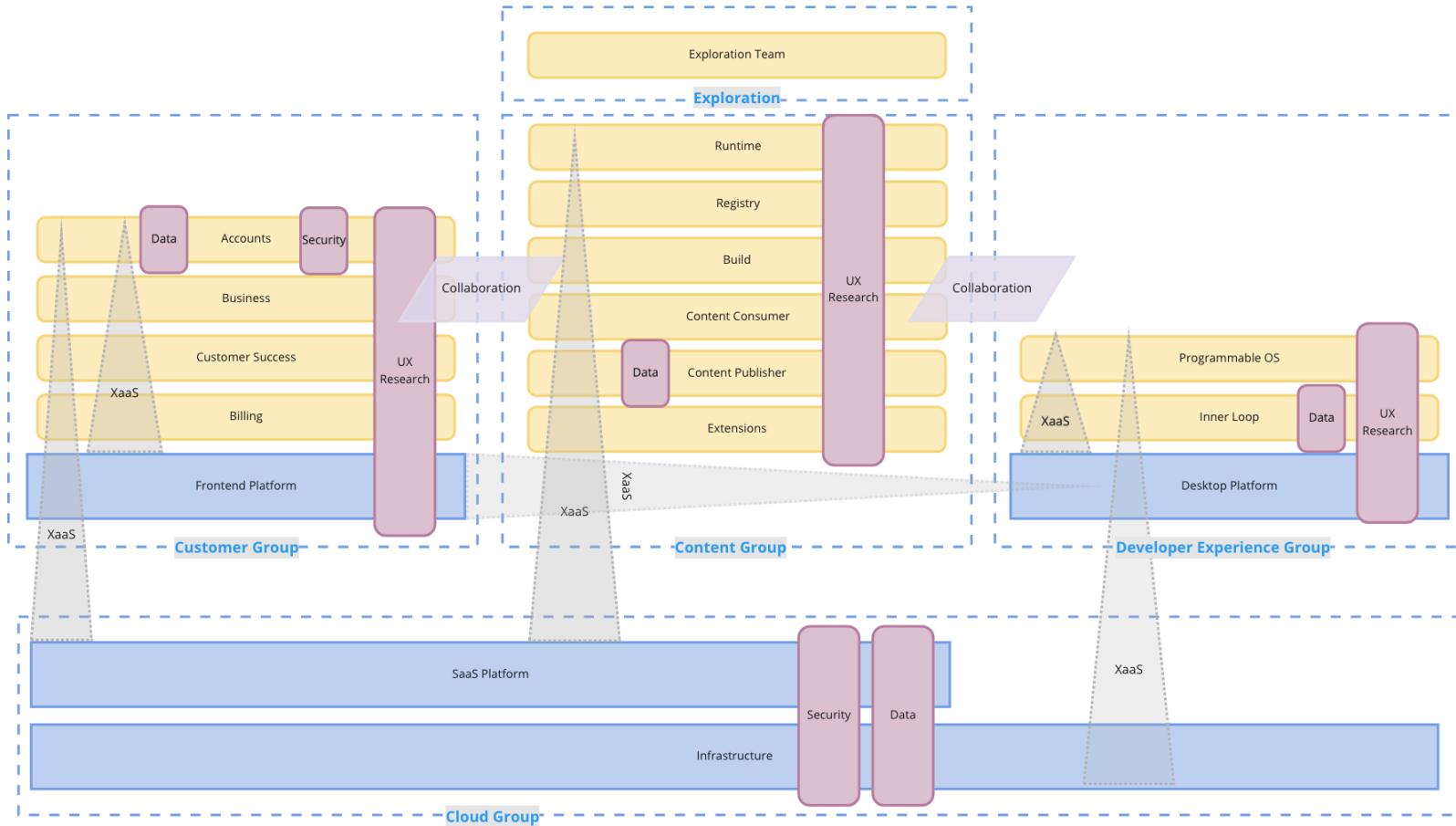
Fin 2019 – 2021

~7

Mi 2022

23+





Dockercon: Team Topologies Methodologies While Renavigating and Rescaling the Industry Standard for Containerization

🎥 <https://docker.events.cube365.net/dockercon/2022/content/Videos/1adc6649-99cd-4535-880a-1bc02b7483f1>



💬 <https://teamtopologies.com/industry-examples/rebuilding-and-scaling-product-development-at-docker-using-team-topologies>





Communication is

hard

key



La technique aide
à réduire la charge cognitive





Il n'y a pas une bonne solution
mais plusieurs mauvaises

Merci



Questions, remarques, avis



<https://roti.express/r/alp-02>



@_crev_

yves.brissaud@docker.com



Illustrations, ressources

<https://teamtopologies.com/>

<https://github.com/TeamTopologies>

<https://unsplash.com/photos/r3ply-3Xgmg>

<https://unsplash.com/photos/634KBk3AXNA>

<https://unsplash.com/photos/7kSnMLGoR9w>

<https://unsplash.com/photos/cWcuKut1RAE>

<https://unsplash.com/photos/LJhXYHxPfEY>

<https://telegram.org/blog/stickers-meet-art-and-history>

