Responsibly Designing Open Science Data Teams

Amanda Casari
Google

IEEE OSDC Workshop
December 9, 2019

Google Open Source

🌊 hi, i'm amanda 🌊

Engineering Manager @ Google

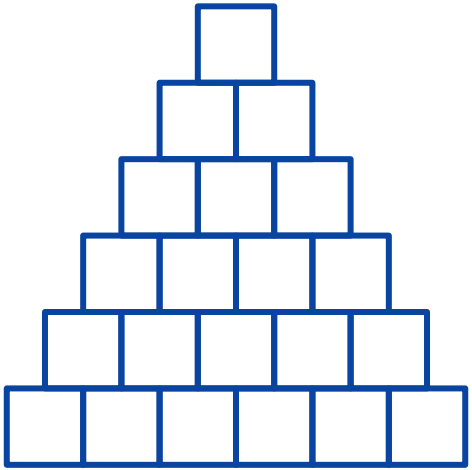
Advisor for Open Source Community Africa

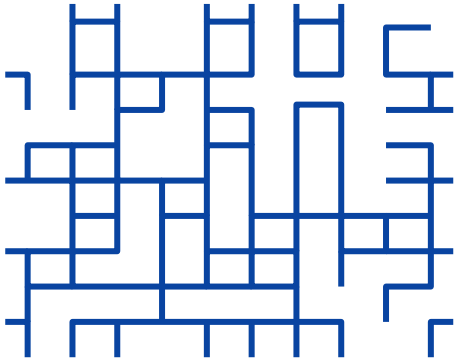
Academic Advisory Board, UBC Master of Data Science Program

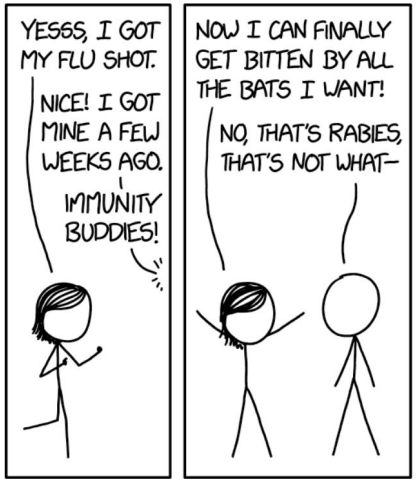
Co-author with Alice Zheng, "Feature Engineering for Machine Learning" [O'Reilly, 2018]

Organizer @ Seattle PyLadies

I am *endlessly fascinated* by the difference between ***the systems we aim to create*** + the ***real systems which emerge***.

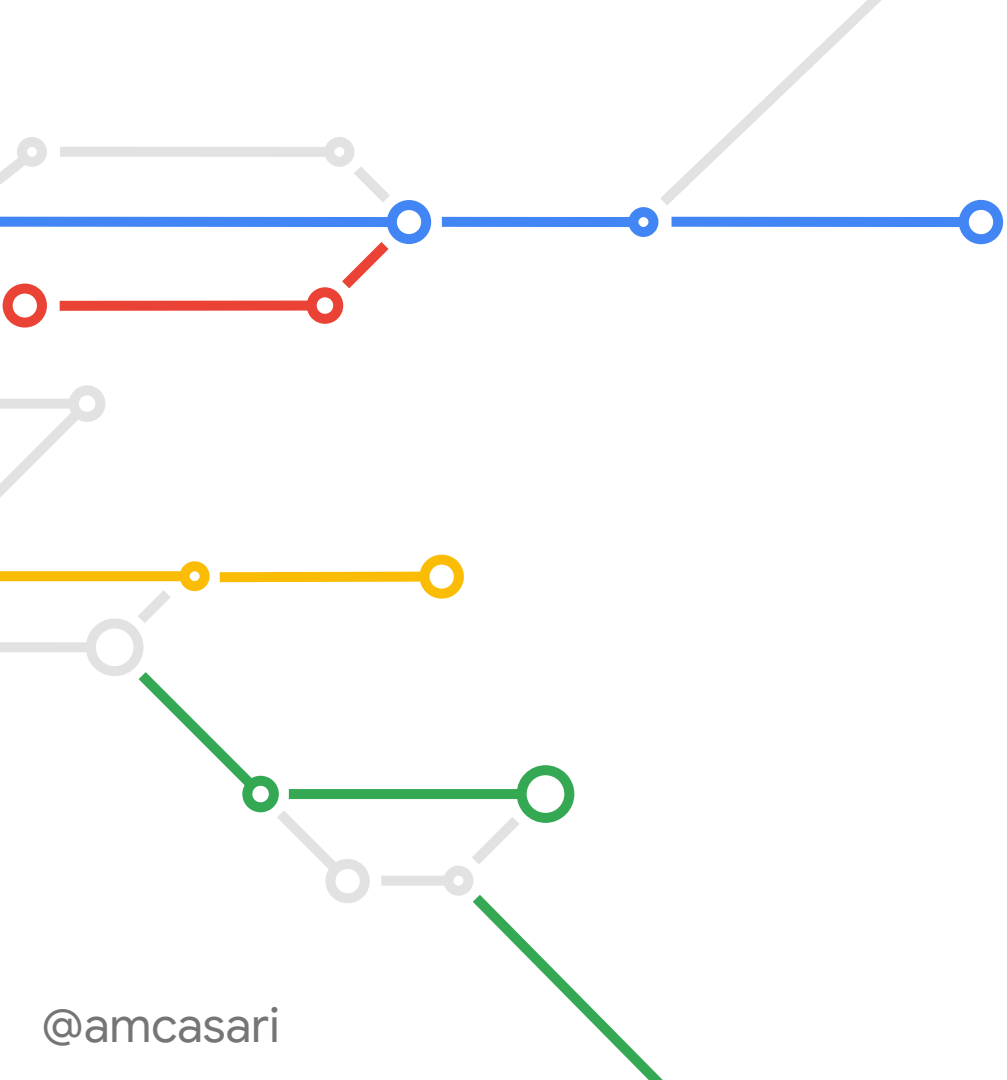






xkcd - flu shot

The Systems We Created + Their Emergent Challenges



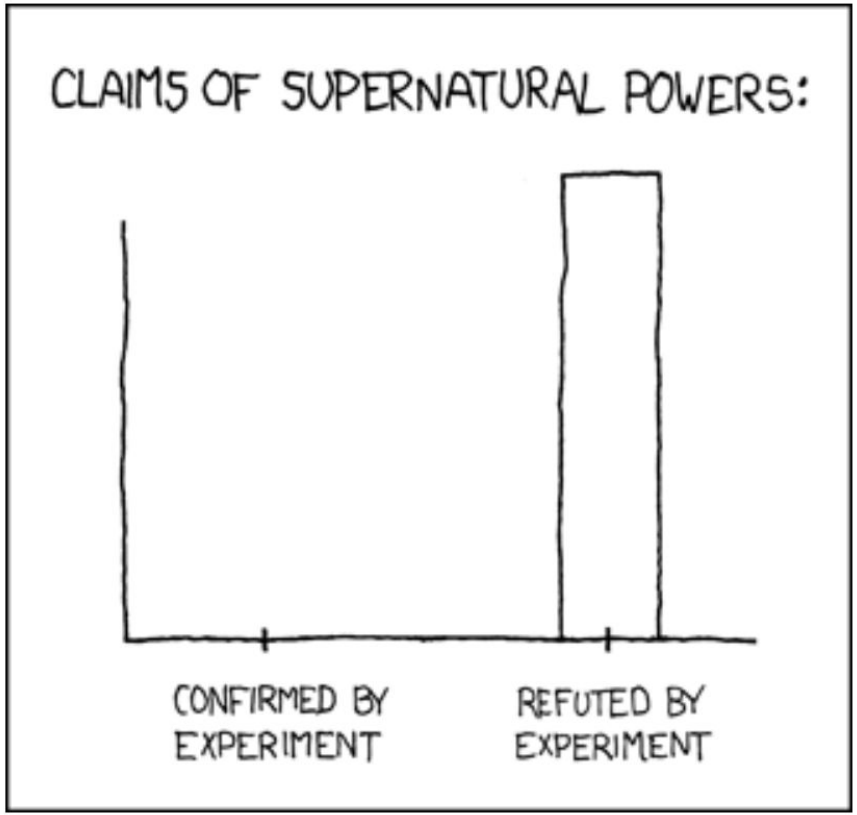
Big Data

The Promise of Big Data

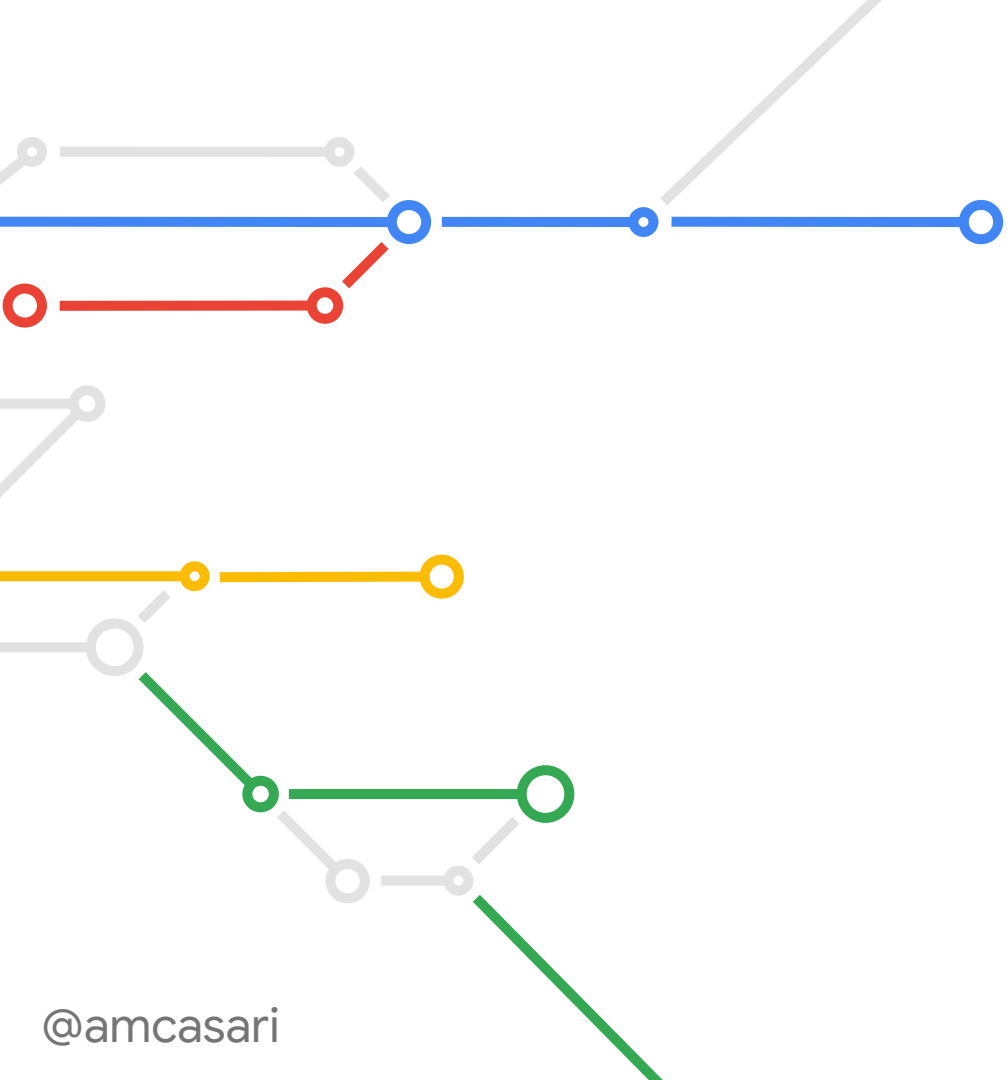
@amcasari

Google Open Source

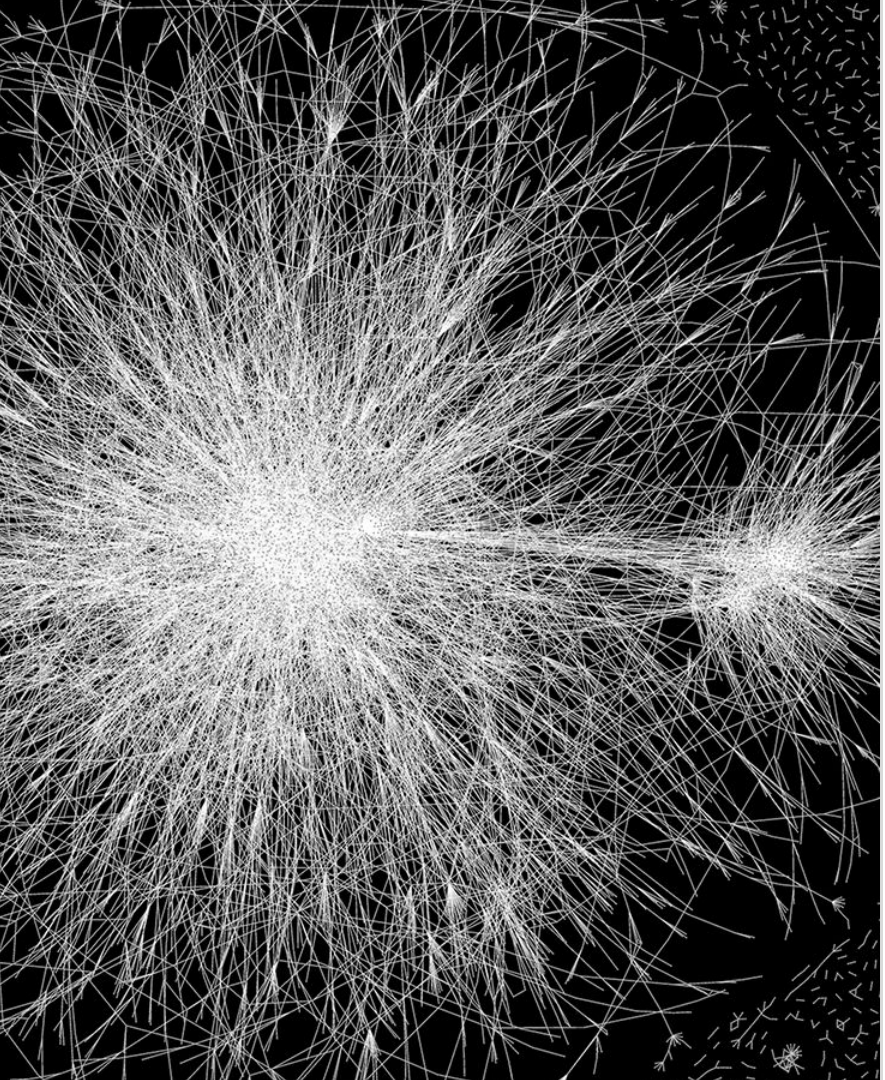
The Reality of Big Data



[xkcd - the data so far]



Data Science

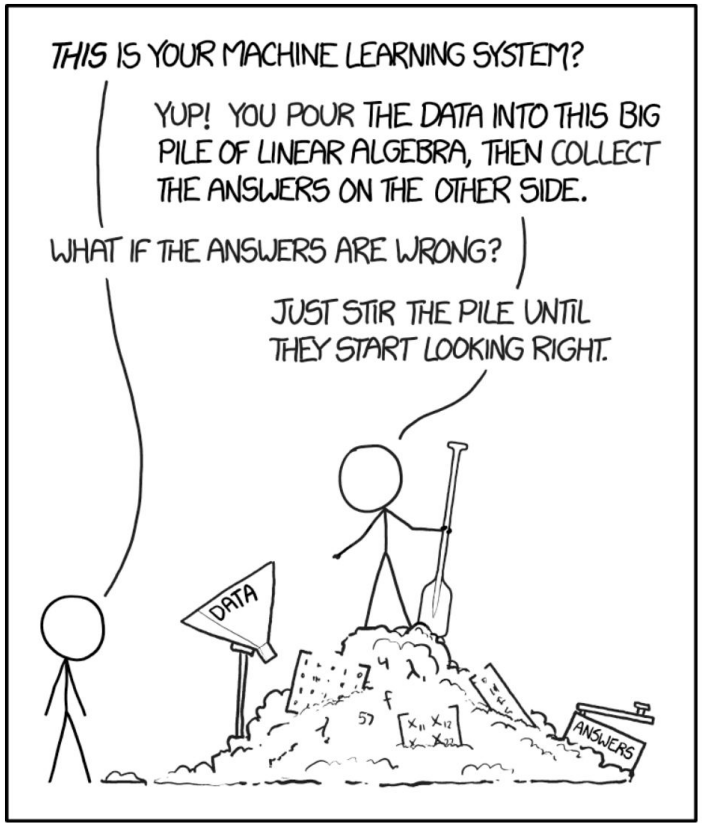


The Promise of Data Science

image: shared channels @ slack

@amcasari

The Reality of Data Science



[xkcd - machine learning]

The Reality of Data Science

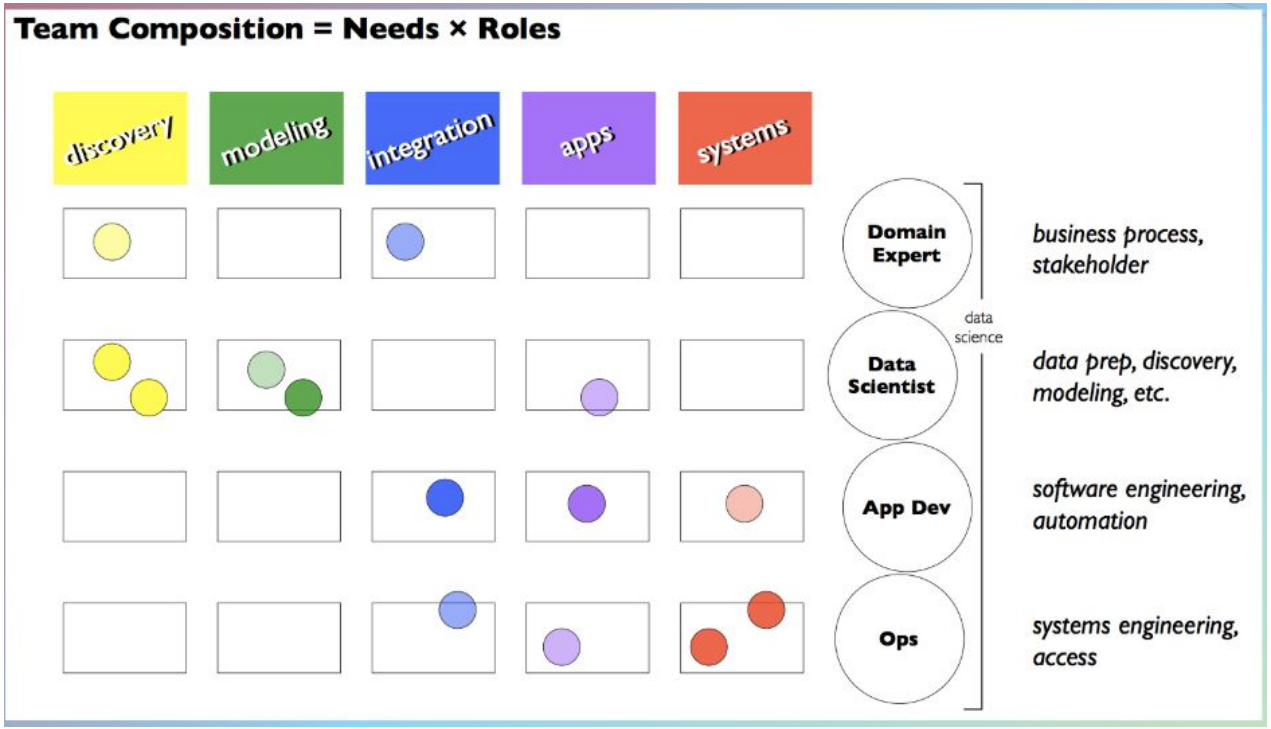
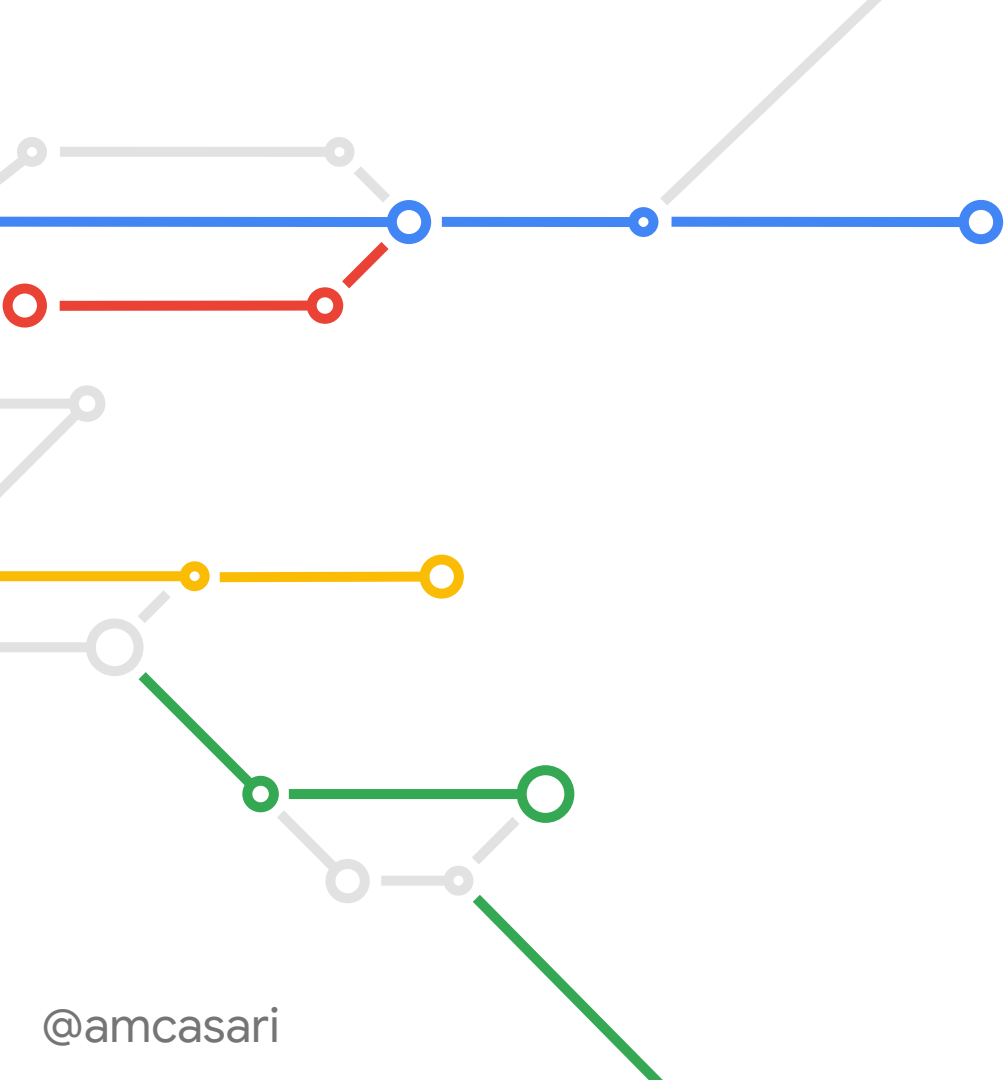


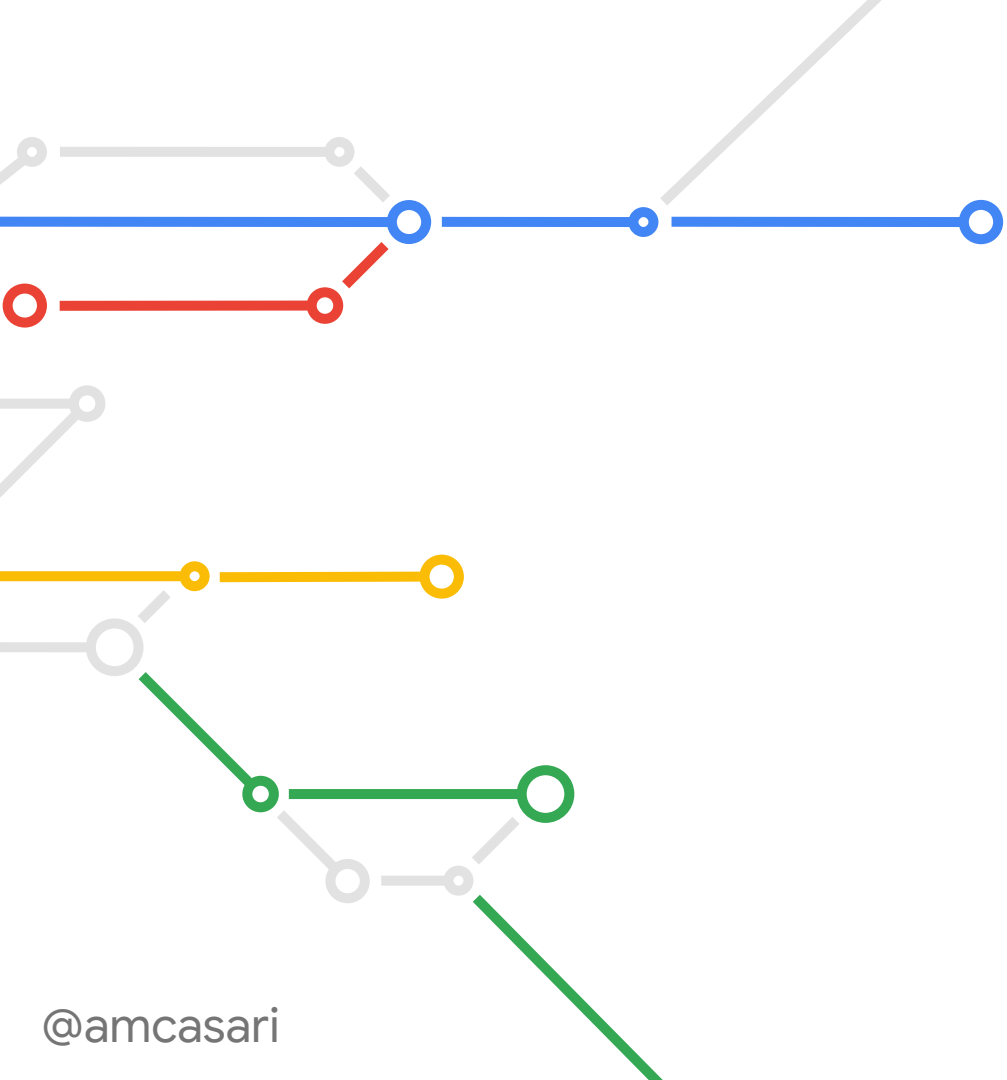
diagram and description courtesy of Paco Nathan

"blurring these roles is wonderful... however, when businesses get into trouble, they also tend to "push down" these roles, **blurring boundaries** in ways which stresses teams and **limits scalability.**

NB: effective, hands-on management in Data Science must live in the space of **integration**, not delegate it"



...sound familiar?

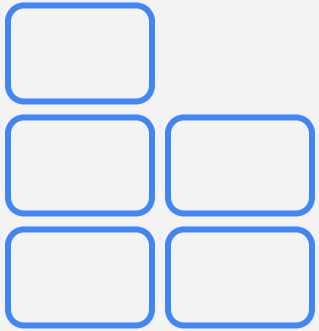


Open Science

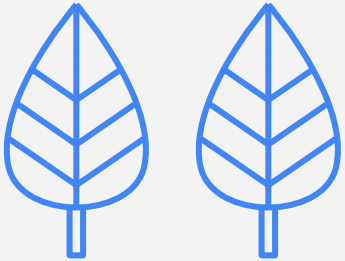
The Promise of Open Science

Resiliency

Scalability



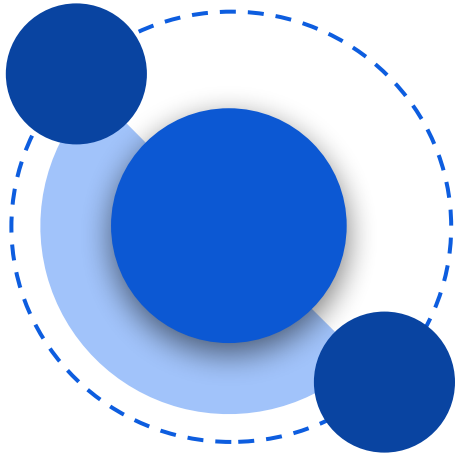
Reproducibility

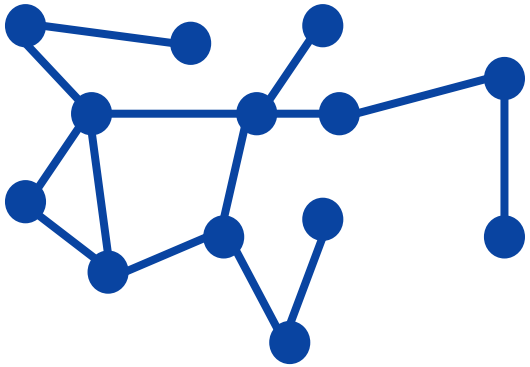


Transparency

...for equity + equality of knowledge.

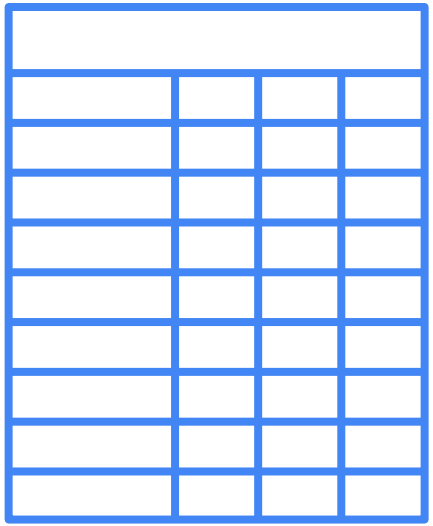
It's about the ***Problems*** + ***Stacks*** + ***Teams***

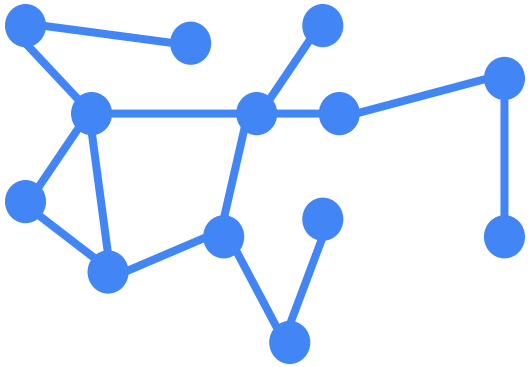




...it's about the interdependent system.

Examine the shape of the system...






Lists as Problems as Stacks as Teams

Lists as Problems...

- Simple dimensions
- Well scoped outcomes
- Success is getting from A to B
- Clear timelines

Lists as Problems as Stacks...

- Use known technical architectures
- Open source for the win!
- Keep it simple
- Success is it can be done 1 to N times well + predictably
- Maintenance is clear

Lists as Problems as Stacks as Teams

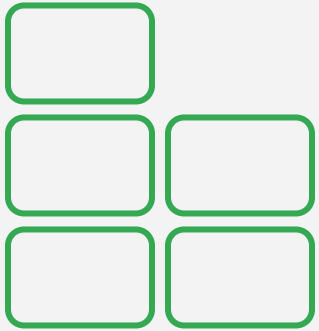
- Role descriptions look much like requirements list
- “Full stack” problem solvers
- Independent individuals
- Growth is methodically extending the project’s known feature set

Lists as... - Managing Your Tradeoffs

Resiliency

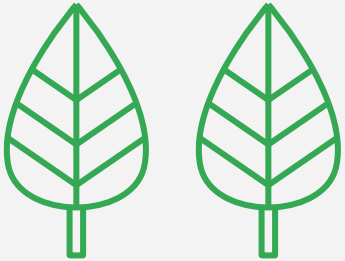
Linear

Scalability



ACID is fine

Reproducibility



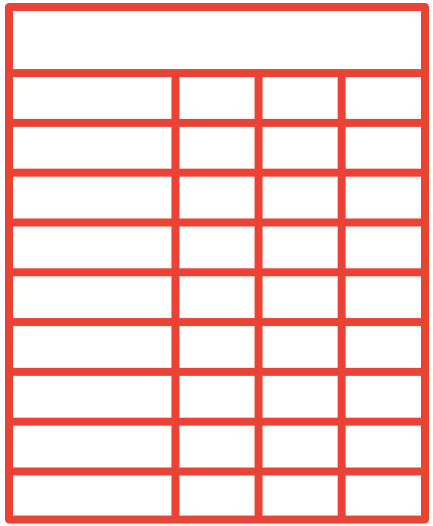
Deterministic

Transparency

Clear

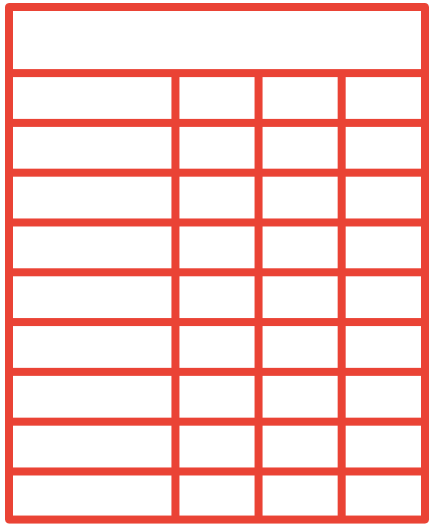
Matrices as Problems as Stacks as Teams

Matrices as Problems...



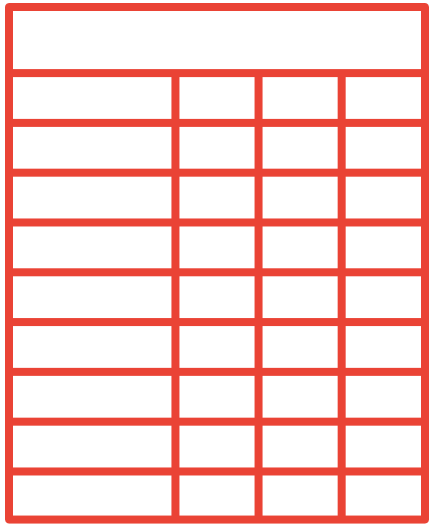
- Problem extends broadly + deeply
- Problem space + outcomes are ambiguous, but constrained
- Timelines can be known, with scoping

Matrices as Problems as Stacks...



- Use known technical architectures
- Centralized versus decentralized versus service versus platform
- Success is it can be done N times repeatably and reasonably fast
- Maintenance is clear + distributed

Matrices as Problems as Stacks as Teams



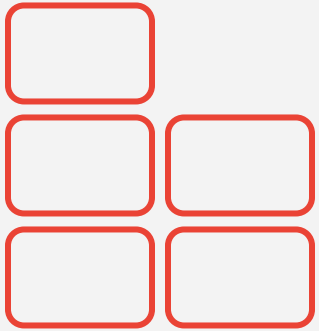
- Roles + skill sets can be distributed across team
- Workloads can spread out
- Known shape of roles + skills needed for success
- Fixed mindsets work well here - lots of knowns, clear expectations

Matrices as... - Managing Tradeoffs

Resiliency

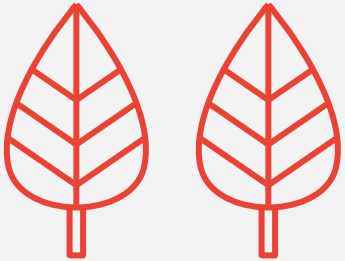
Big O

Scalability



CAP applies

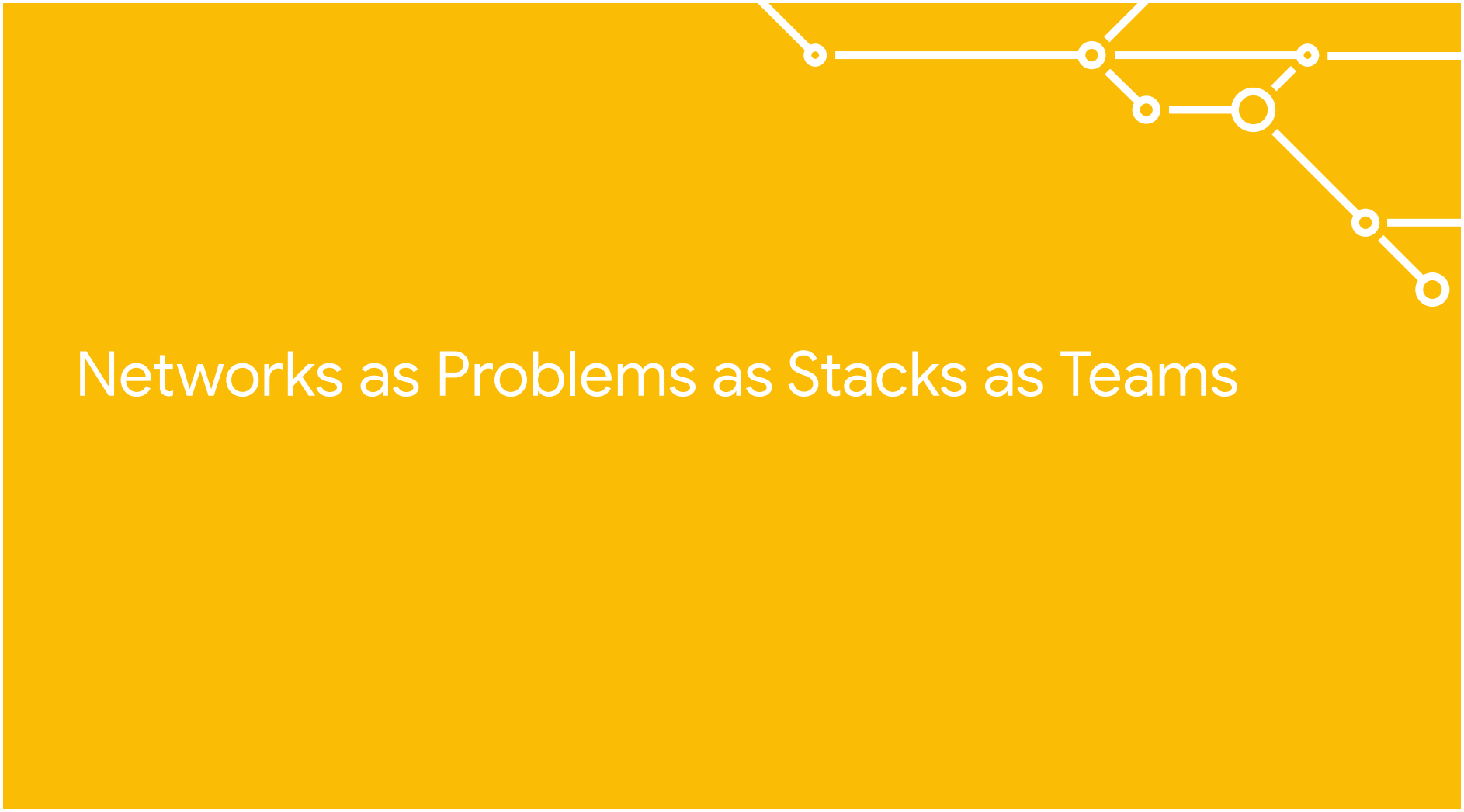
Reproducibility



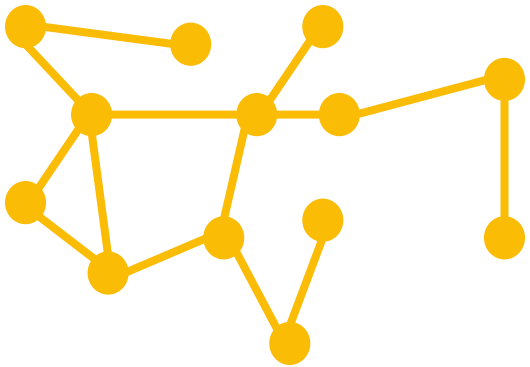
Deterministic

Transparency

Distributed

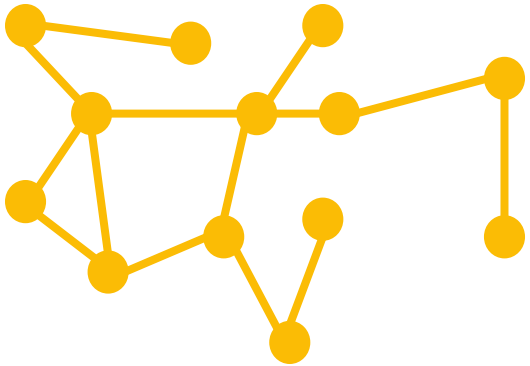

Networks as Problems as Stacks as Teams

Networks as Problems...



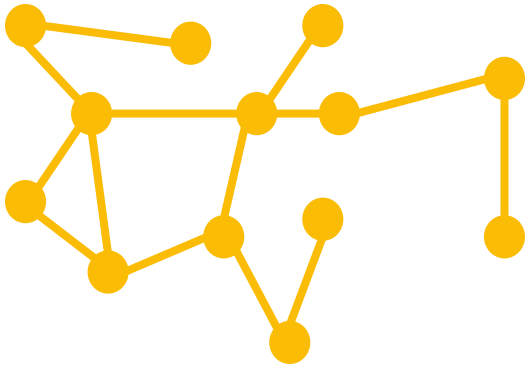
- Problem space is ambiguous
- Constraints can change over time
- Clear short term timelines + outcomes
- Clear long term vision + goals
- Path between short + long term is not necessarily developed yet

Networks as Problems as Stacks...



- Plan for growth + modularity
- Open source for the win!
- Expect needs to change over time
- Expect complexity to grow over time
- It’s not done until it’s documented

Networks as Problems as Stacks as Teams



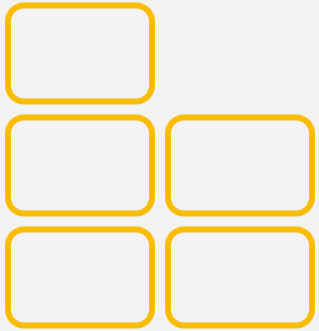
- Nodes
  - Distribution of skills matters
  - Adjacency of roles/skills can flex
- Edges
  - Communication is critical
  - It's not done until it's documented + open to the team + accessible

Networks as... - Managing Tradeoffs

Resiliency

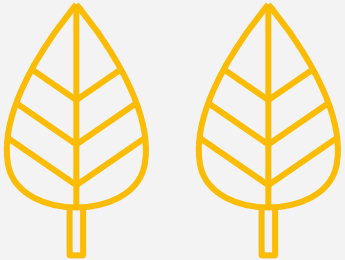
Nonlinear

Scalability



CAP applies

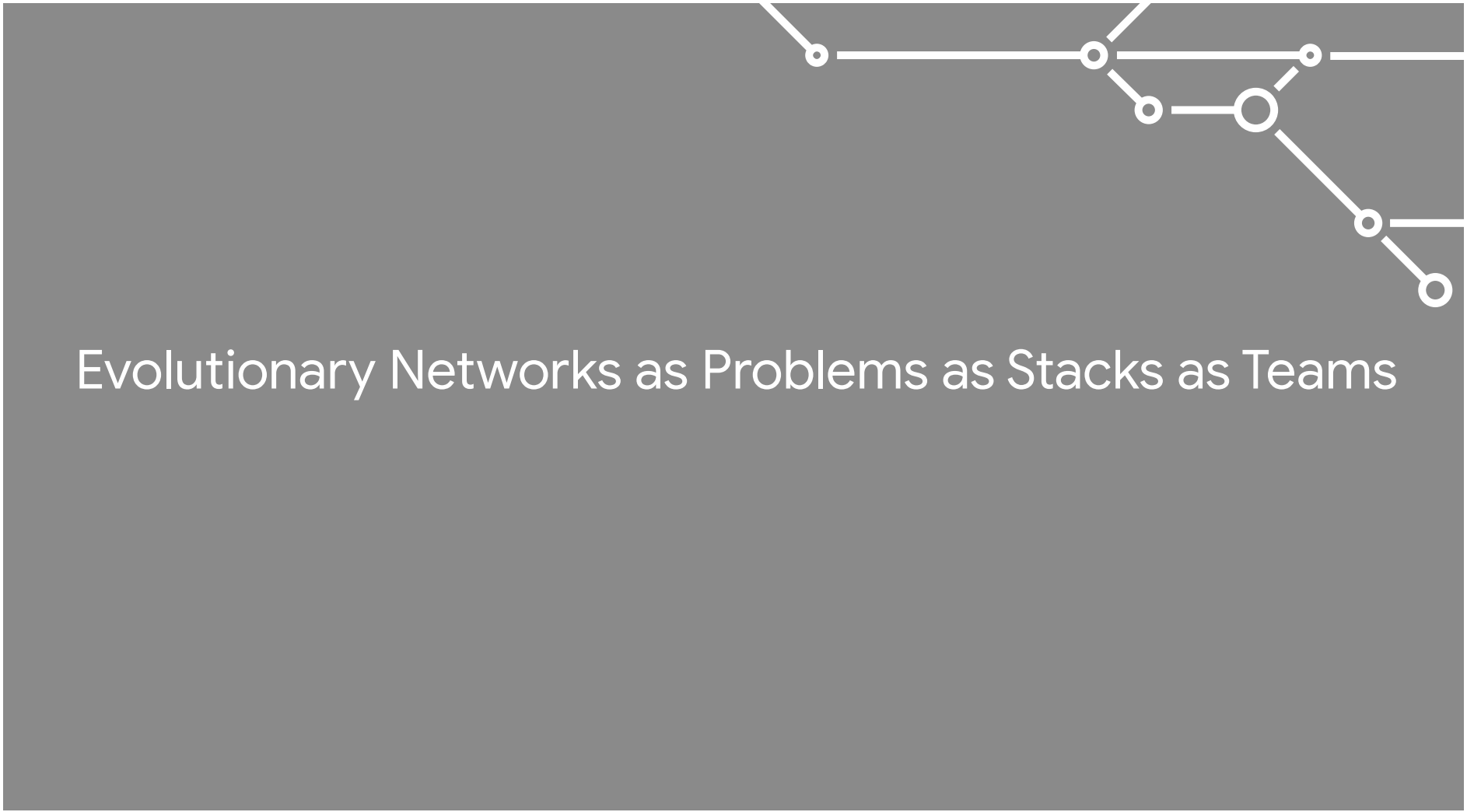
Reproducibility



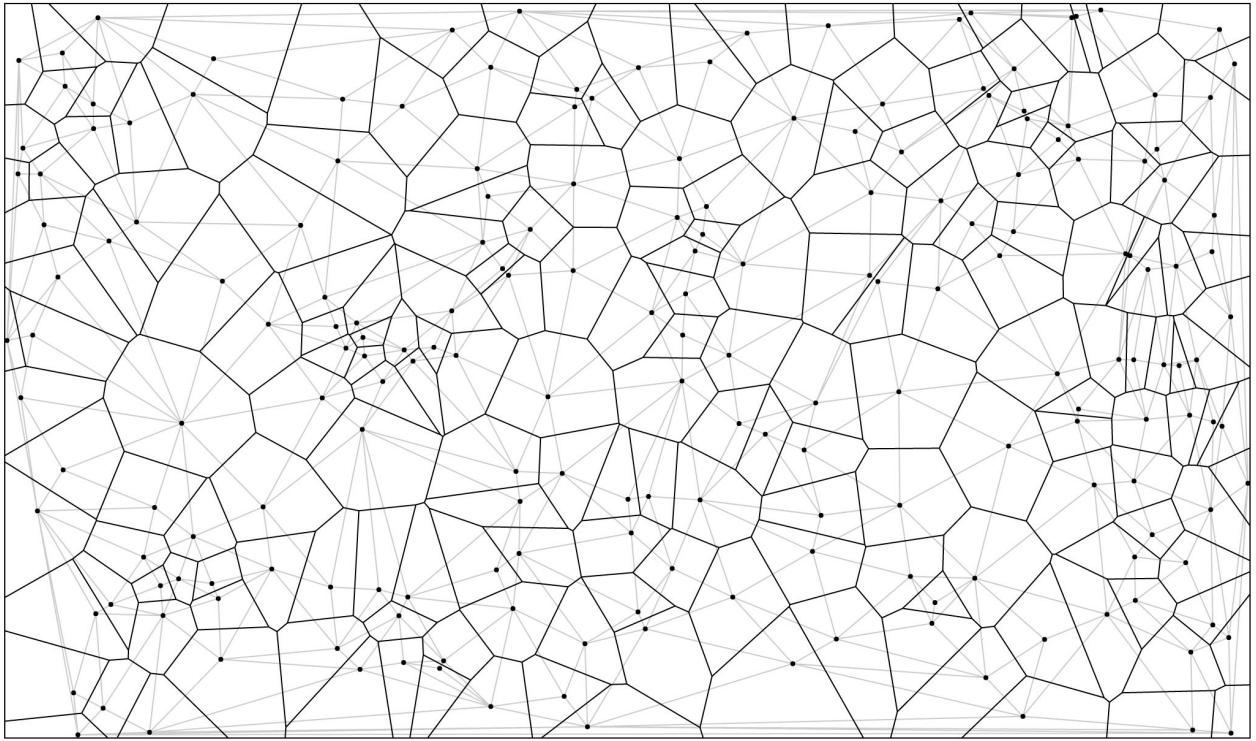
Probabilistic

Transparency

Observable

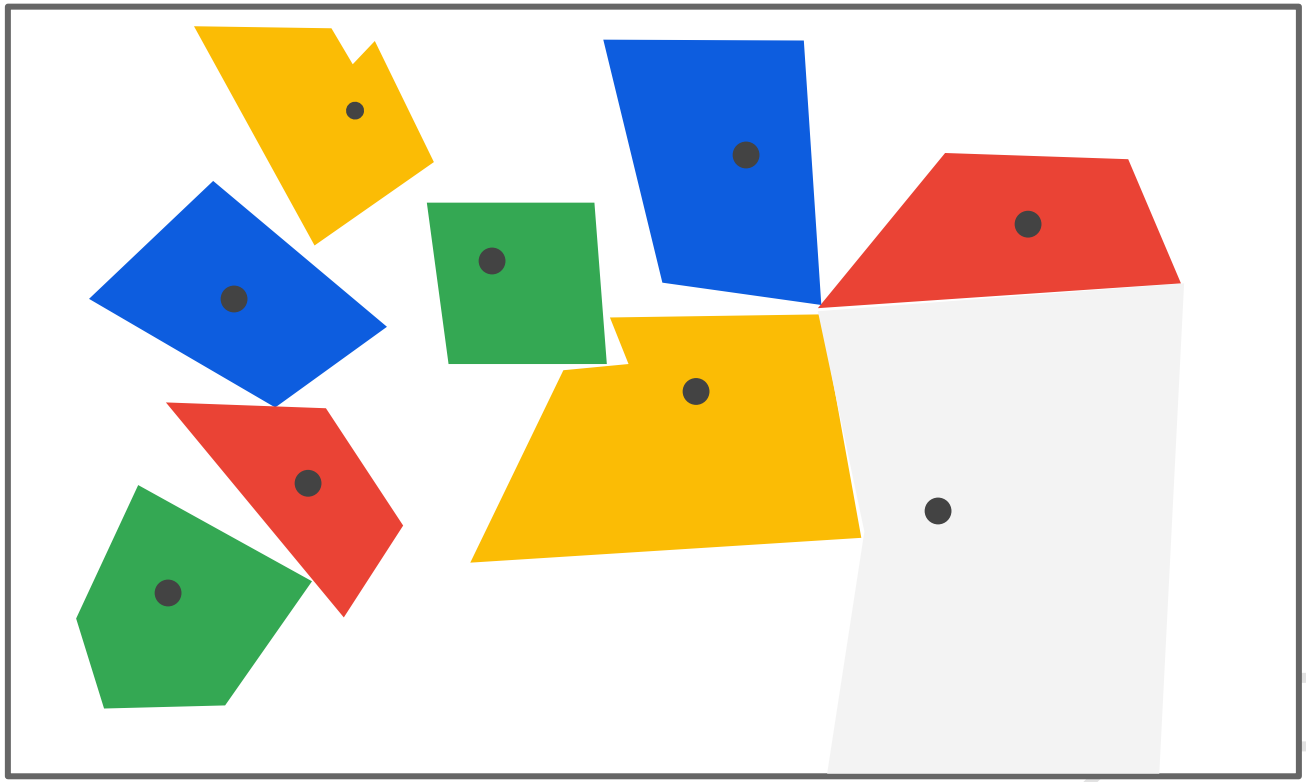

Evolutionary Networks as Problems as Stacks as Teams

Evolutionary Networks as... Space Filling Shapes



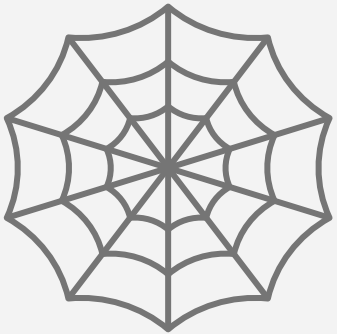
@mbostock - voronoi

Evolutionary Networks as... Room to Grow



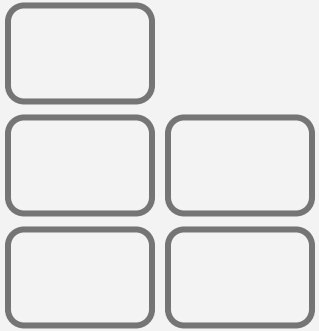
Evolutionary Networks - Managing Tradeoffs

Resiliency



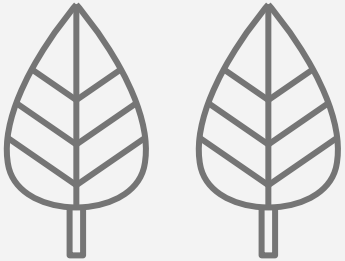
Nonlinear

Scalability



CAP applies

Reproducibility

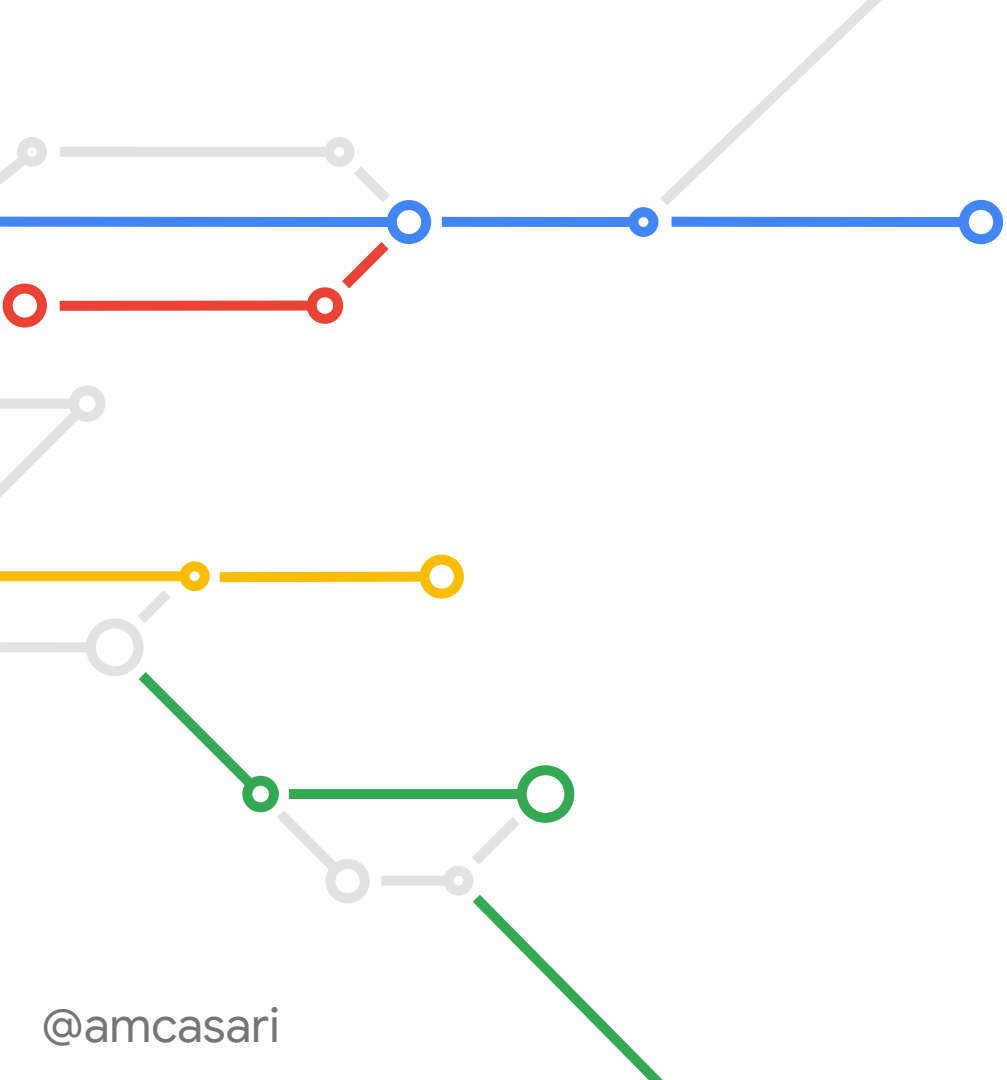


Probabilistic

Transparency

Observable

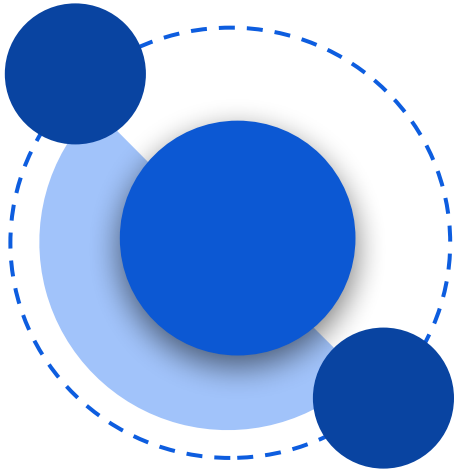
...same as other networks
(with potentially greater return on team investment)

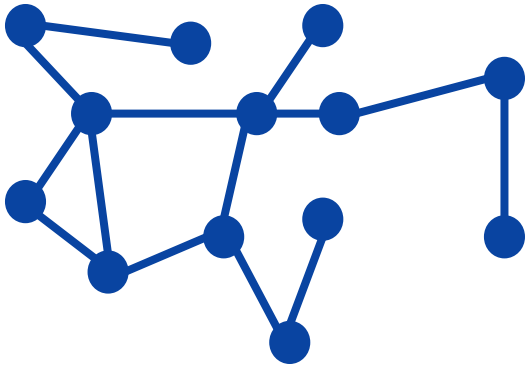


The Path Forward

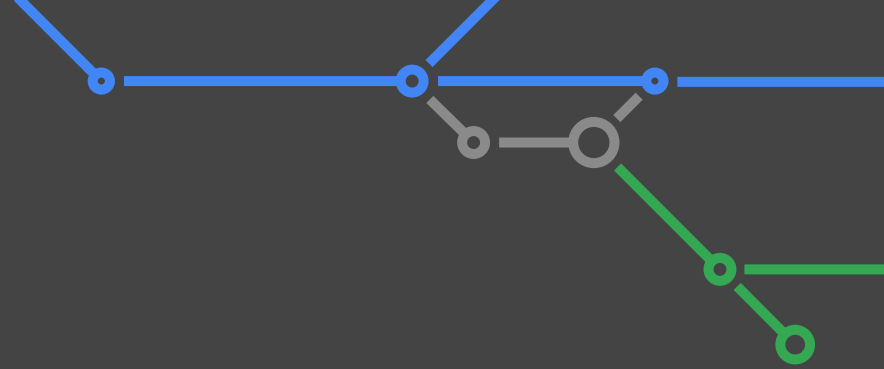
To design systems responsibly --

- Find the shape of your problem
- Consider your (necessary) constraints
- Consider your resources: technical + people
- Consider your tradeoffs
- Find where you can leave room for growth





Design your projects responsibly, as interdependent systems:
people + tech + resources + constraints



Thank you

Amanda Casari

@amcasari

opensource.google.com

Recommended reading on building teams + complexity

- *The Girls of Summer: The U.S. Women's Soccer Team and How It Changed the World*, Jere Longman (2001)
- *The Manager's Path: A Guide for Tech Leaders Navigating Growth + Change*, Camille Fourier (2017)
- *Resilient Management*, Lara Callender Hogan (2019)
- *Systems Thinking, A Primer*, Donella Meadows (2008)
- *An Elegant Puzzle: Systems of Engineering Management*, William Larson (2019)