28 Sept. 2021

Bertha

- portability (apps to networks)
- using changing set of network services
- picking right implementation

CCP

- implementations (algs to datapaths)
- increasingly complex algs need to fit in restrictive datapath envs.

these are not problems if you have lots of resources:

- implement compatibility for everything
- enforce standardized operating environment
- economics of scale: chase the last bit of perf.

⇒ how can we democratize applications' access to modern networks?.

↳ networks are programmable, but require too much effort to actually program

↳ programmability unlocks applications' use of modern network features

↳ alternative: API lock-in from cloud providers

14 Oct Mohammad feedback

- story is a little too complicated
  - ↳ simple nugget explainable in 2-3 mins
  - ↳ novelty/impact + techniques
- setting up network stack
  - ↳ start with simple 3 sentence version and minimally expand
    - → still think about 1-sentence statement: "abstractions that help apps deal w/ net complexity" is a bit vague: want a crisper image
    - ↳ look at Keith's story: need something easy to understand

→ apps have sidestepped the traditional net. stack, ↳ "dumb pipes" instead using custom libs. in a piecemeal way.

→ Bertha doesn't itself impl. services, it gives apps a way to express what they want

↳ so the democratization angle is more about unusual compositions

↳ attempting to desc, some ways are very high level, other ways focus on very specific examples. need a middle ground explanation: ad-hoc interfaces → composability + portability + exactness. "trends I'm concerned about"

complex apps →
network heterogeneity ↗ adhoc interfaces → 2 trends won't work out
↳ new abstractions for apps to express what they actually want

⇒ are there short statements that capture the essence of the work? (architecture angle will stand out)

new arch for modern apps that allow them to take adv. of new trends in networks

↳ second level: what makes me uniquely qualified

↳ what is the bigger message

1. "Apps rule the world" + are very important in themselves ⇒ networked (11pt font)
2. apps have become complex
   - ↳ compose features, require sec. + avail + perf + scalability due to high importance ⇒ result is complexity
   - ↳ wide range of net. services fore sec. rel., perf. ⇒ embed libraries. which compose in unexpected ways, and ...

→ make devs more productive

constituency (apps) → devs → how are they getting services they need

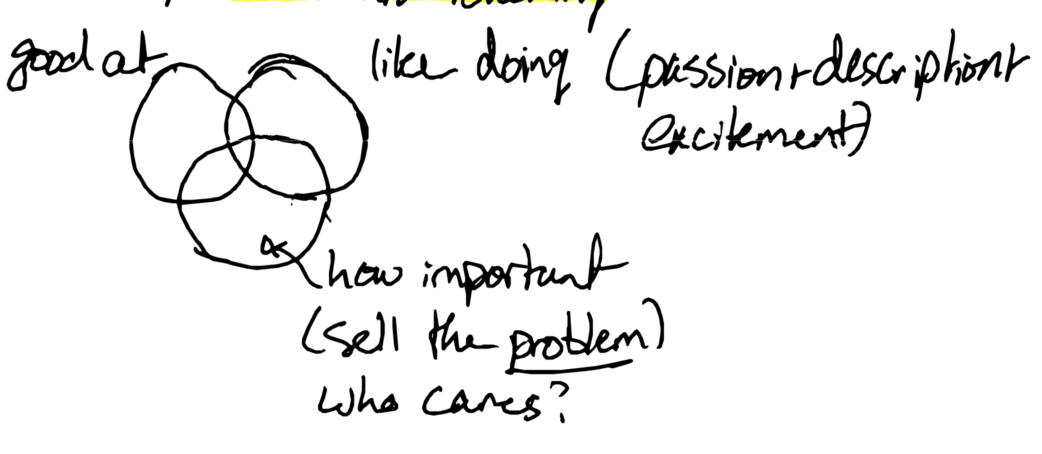
→ "ad hoc" is not useful

→ make CCP be a use case

→ don't need to explain solution in the first part, only problem statement

→ need mission statement

→ tie composition into teaching


good at
like doing (passion + description + excitement)
how important (sell the problem) who cares?

Hari: 9 March 2022

- why not have the server send a script to the client? (bertha)
  - ↳ performance ↔ flexibility tradeoff: we don't actually want to have this script thing because we would have to emted <u>all</u> functionality inside which is ↓ perf, instead w/ Bertha we can just switch between existing stacks which gets us a good level of both flex. and perf.
- extending Bertha for edge computing (future work)
- explain what the difference is between Bertha and a wrapper library (in terms of portability, etc) Similarly, what is the difference between Bertha and a proxy/tunnel for composability
  - ↳ motivate Bertha components and Bertha by providing a list of required features.
- intro: new trends in apps, in technology, and in users: apps using comm. libs., tech for new high perf datapaths, and users demanding performance and <u>flexibility</u>.