



oueees-201706

Part 1: Tragedy of sharing

Kenji Rikitake

13-JUN-2017

School of Engineering Science

Osaka University

Toyonaka, Osaka, Japan

@jj1bdx

Lecture notes on GitHub

- <https://github.com/jj1bdx/oueees-201706-public/>
- Don't forget to *check out the issues!*

A close-up photograph of a pair of hands holding a shallow wooden bowl filled with ripe, red cherry tomatoes. The hands are positioned on either side of the bowl, with the fingers gently gripping its rim. In the background, a burlap bag with a blue polka-dot pattern is visible, resting on a grassy surface. The overall scene conveys a sense of freshness and sharing.

Sharing

Share (v.)¹

Have/give a portion of (something) with another or others

Use, occupy, or enjoy (something) jointly with another or others

Possess (a view or quality) in common with others

¹ New Oxford American Dictionary, macOS 10.12.5

More on share (v.)²

Tell someone about
(something), *especially*
something personal

**Post or repost (something)
on a social media website or
application**

² New Oxford American Dictionary, macOS 10.12.5,
emphasis by Kenji Rikitake



Purposes of sharing: Showing off and Saving resources

Showing off

- Publicly bragging
- Addiction for Approval
- 「私かわいい」
- ... and various other psychological reasons
- Creating a lot of social problems
- To be discussed in Part 3



Sharing on programming

Historical background on computing resources

- CPU speed
- Memory
- Storage
- Network bandwidth
- ...All always on shortage

Sharing, programming and memory

- Memory is expensive: *explicit* allocation required
- Variables are *mutable*
- Internal state is *commonly shared* and accessible between multiple functions and modules
- Use *memory pointers* to minimize the number of copying, inherently suggesting: *share as much as you can*

**Question: is
sharing
*intuitive?***

In JavaScript (node.js)

```
// var a = {first: 1, second: 2}
// b = a // only share pointer
{ first: 1, second: 2 }
// a.second = 3
3
// b // element is shared
{ first: 1, second: 3 }
// b == { first: 1, second: 3 }
false // WHY?
```

Copying *by default, not* sharing, can solve this issue

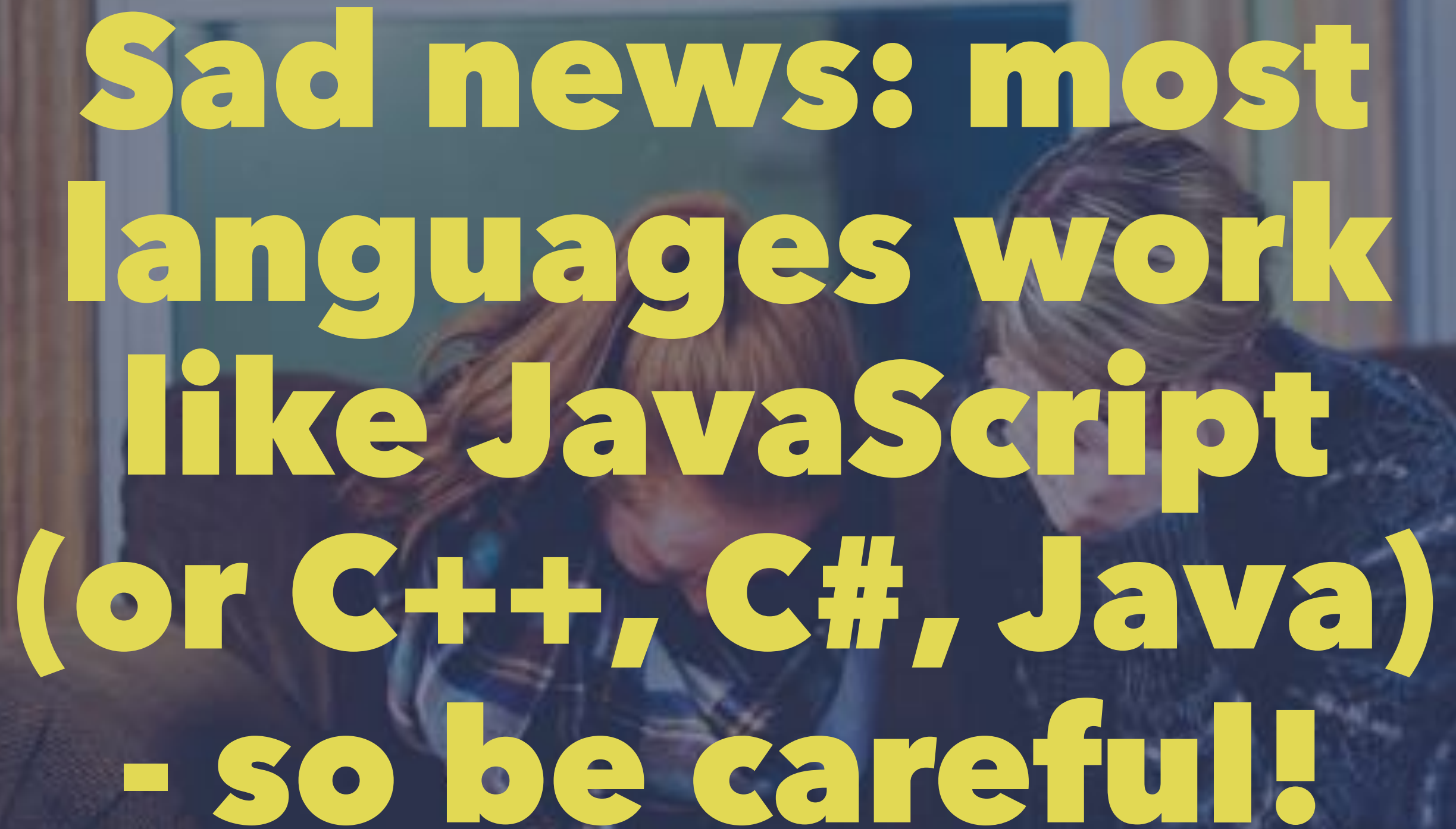
In Elixir (v1.4.4)

```
iex(1)> a = %{first: 1, second: 2}
%{first: 1, second: 2}
iex(2)> b = a # copying the map
%{first: 1, second: 2}
iex(3)> a = %{a | second: 3}
%{first: 1, second: 3} # member modified
iex(4)> b # not shared with a
%{first: 1, second: 2}
iex(5)> b == %{first: 1, second: 2}
true # intuitive!
```

**Thought: sharing
is *not necessarily*
*intuitive***

Issues of sharing-based programming languages

- Access violation between multiple programs
- Zombie memory area without ownership
- Need for explicit copying cause bugs
- Mutable states are difficult to debug
- Semantically sharing is a *shortcut* and breaks many logical assumptions

A background image showing two people, a man and a woman, sitting at a desk and looking at a laptop screen. The image is dimmed and serves as a backdrop for the text.

**Sad news: most
languages work
like JavaScript
(or C++, C#, Java)
- so be careful!**



False assumptions on sharing over networks

1-4 of Eight Fallacies of Distributed Computing³

- The network is reliable
- Latency is zero
- Bandwidth is infinite
- The network is secure

³ <https://blog.fogcreek.com/eight-fallacies-of-distributed-computing-tech-talk/>

5-8 of Eight Fallacies of Distributed Computing⁴

- Topology doesn't change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

⁴ <https://blog.fogcreek.com/eight-fallacies-of-distributed-computing-tech-talk/>

The network is *not* reliable

- Somebody breaks the link (cut the line/fiber)
- Error rate of wireless/radio communication is far higher than the wired communication
- Data over the network *may* be altered without being discovered

Latency is *not* zero

- Speed of light: $\sim 300,000\text{km/s}$
- = only **$\sim 30\text{cm/ns}$** , **$\sim 300\text{km/ms}$**
- Even slower on the optic fiber (**$\sim \times 0.7$**)⁵
- Japan-US West Coast: **$\sim 0.1\text{s}$** for round-trip
- **Light is SLOW**

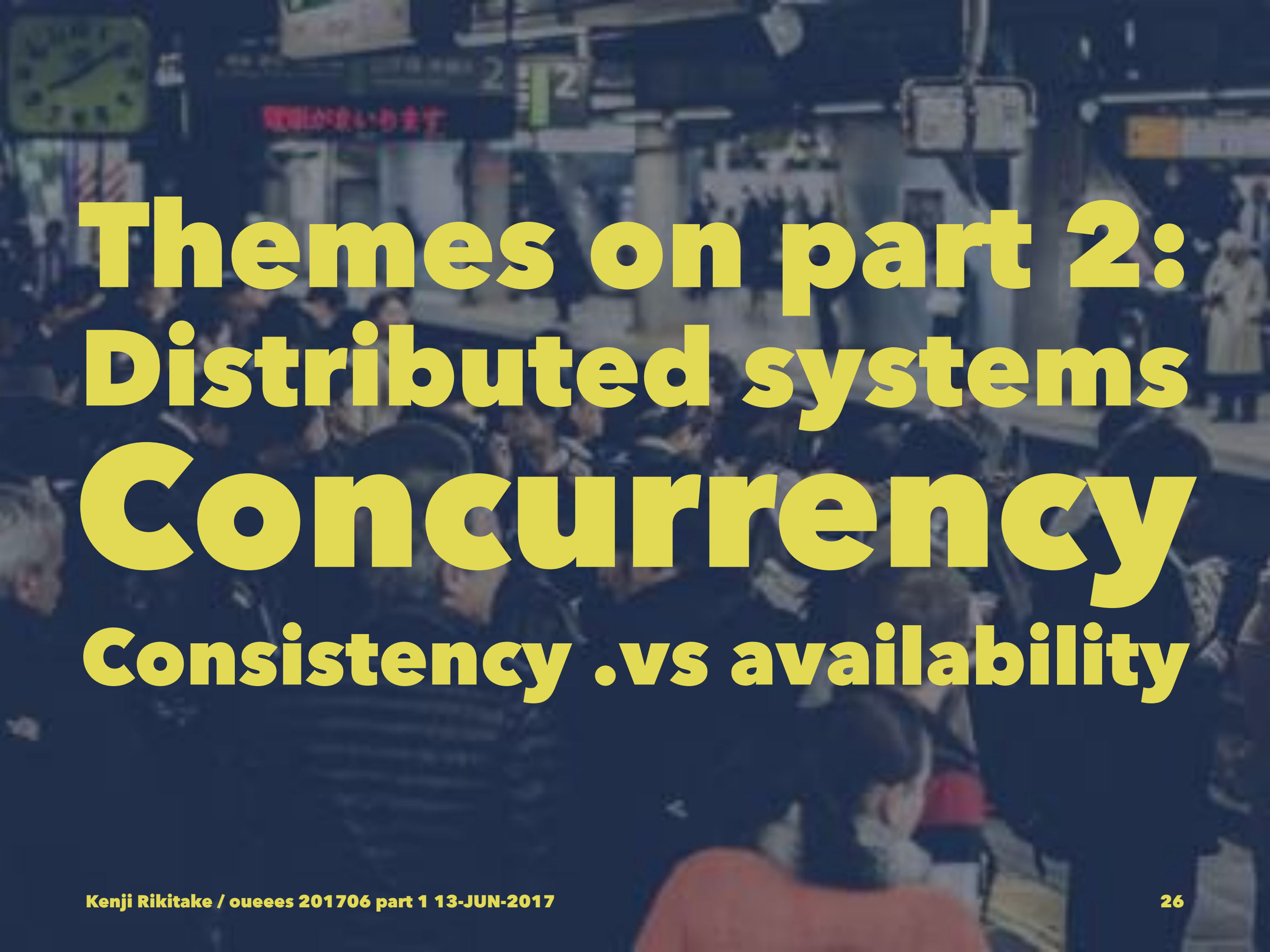
⁵ <https://physics.stackexchange.com/questions/80043/how-fast-does-light-travel-through-a-fibre-optic-cable>

Bandwidth is *limited*

- 1Gbps on Ethernet: ~100Mbytes/sec
- **32Gbytes takes 32 seconds** on Ethernet
- **32Gbytes takes ~54 minutes** on ~1Mbytes/sec link
- **Replication of pictures and videos takes minutes or even hours**

Implications

- What you think you have successfully shared over the network **might be delivered corrupted** or **would not be delivered at all**
- **Sharing might not be completed as you expect**, especially regarding network errors
- **Data delivery delays**

A photograph of a crowded Japanese train station, likely Shinjuku, with many people waiting. The image is overlaid with a semi-transparent blue filter and large yellow text.

Themes on part 2:

Distributed systems

Concurrency

Consistency .vs availability

Photo credits:

- All photos are modified and edited by Kenji Rikitake
- Photos are from Unsplash.com unless otherwise noted
- Title: Alissa Eady
- Sharing: Elaine Casap
- More on share: [Kenji Rikitake from Instagram](#)
- Showing off: Anthony Delanoix
- Sharing on programming: Matthew Henry
- Historical background on computing resources: Damjan Dobrilla
- Sad news: Ben White
- False assumptions on sharing over networks: Fré Sonneveld
- Themes on part 2: Redd Angelo