Post-Collapse

Computing

Tobias Bernard


1.5
R.I.P.

Trigger Warning

Very depressing climate facts


30

PART 1

THE CLIMATE CRISIS IS HERE

Death zones around the equator

Famines (in the global north)

Wars for resources

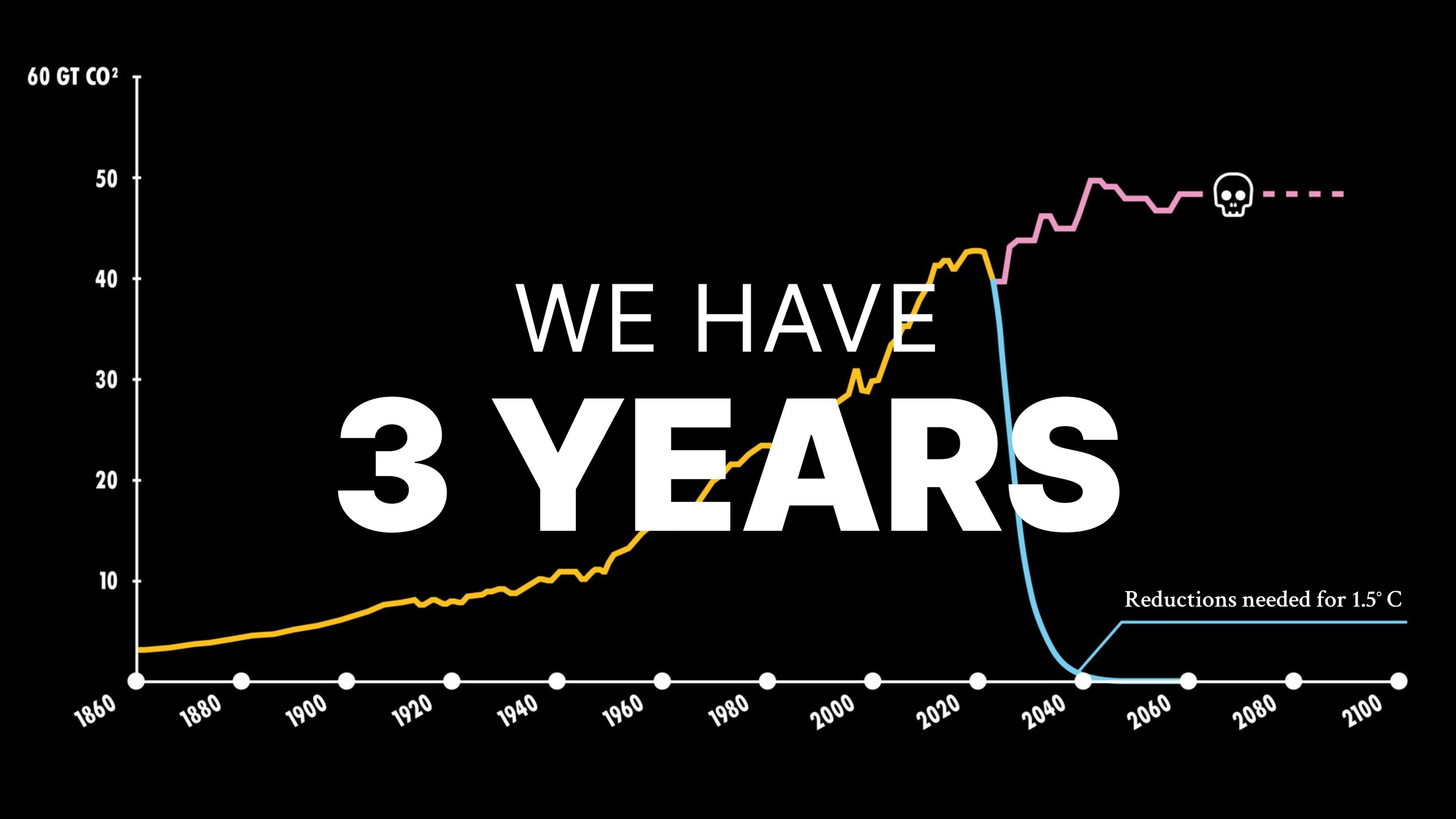
Infrastructure breakdown

Billions of refugees

Creative new takes on fascism

Tipping Points

Feedback Loops

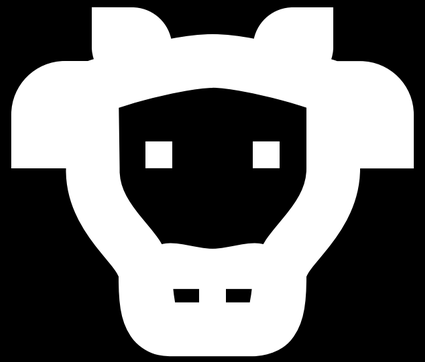

60 GT CO²
50
40
30
20
10
1860
1880
1900
1920
1940
1960
1980
2000
2020
2040
2060
2080
2100
WE HAVE
3 YEARS
Reductions needed for 1.5° C

Three years to do what?

Lifestyle changes are greenwashing*

*Except luxury emissions

If you insist...



Beef

Air Travel

All the real solutions are structural

Our political systems are incapable of action

Mass mobilization

Mass civil disobedience

Block the economy

Citizens’ assemblies

“Should we drop everything and start doing blockades?”

Yes*

* We can still build free software on the side

Civil disobedience isn't as scary as you think*

* Depending on where you live


POLIZEI


POLIZEI
BE
2T230
POLIZEI
BE
2T312
POLIZEI
BE
2T320
REBEL

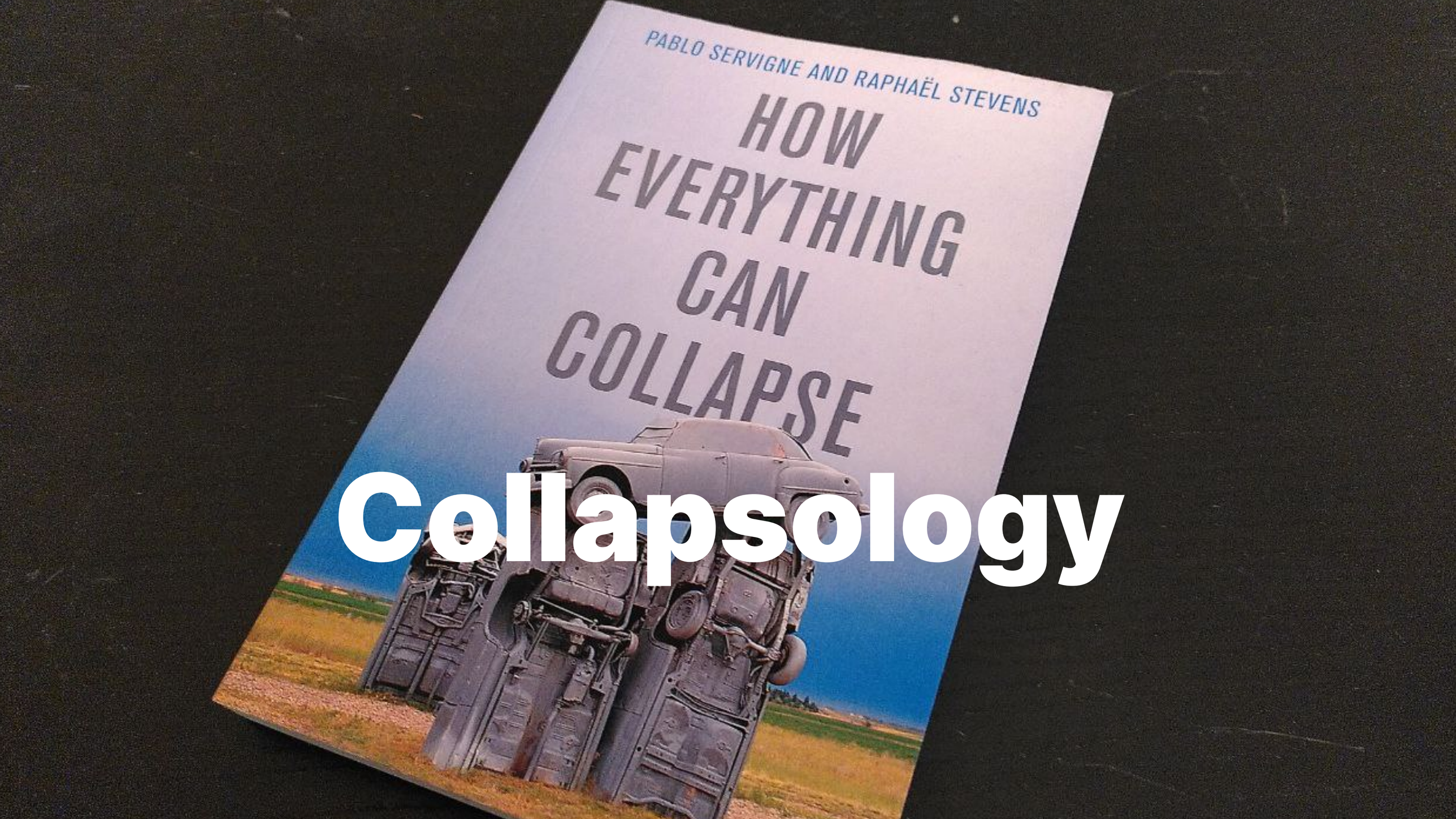

OLIZEI
BE
2T006

Join the rebellion.
It's our best hope.

1 minute


PART 2
WHAT IF WE
FAIL?

What would that mean for GNOME?



Collapsology

What might collapse, degrade, or become inaccessible?

Supply chains

Internet

Power grid

Corporations

(And many other things less relevant to software)

Complex, centralized, interdependent systems

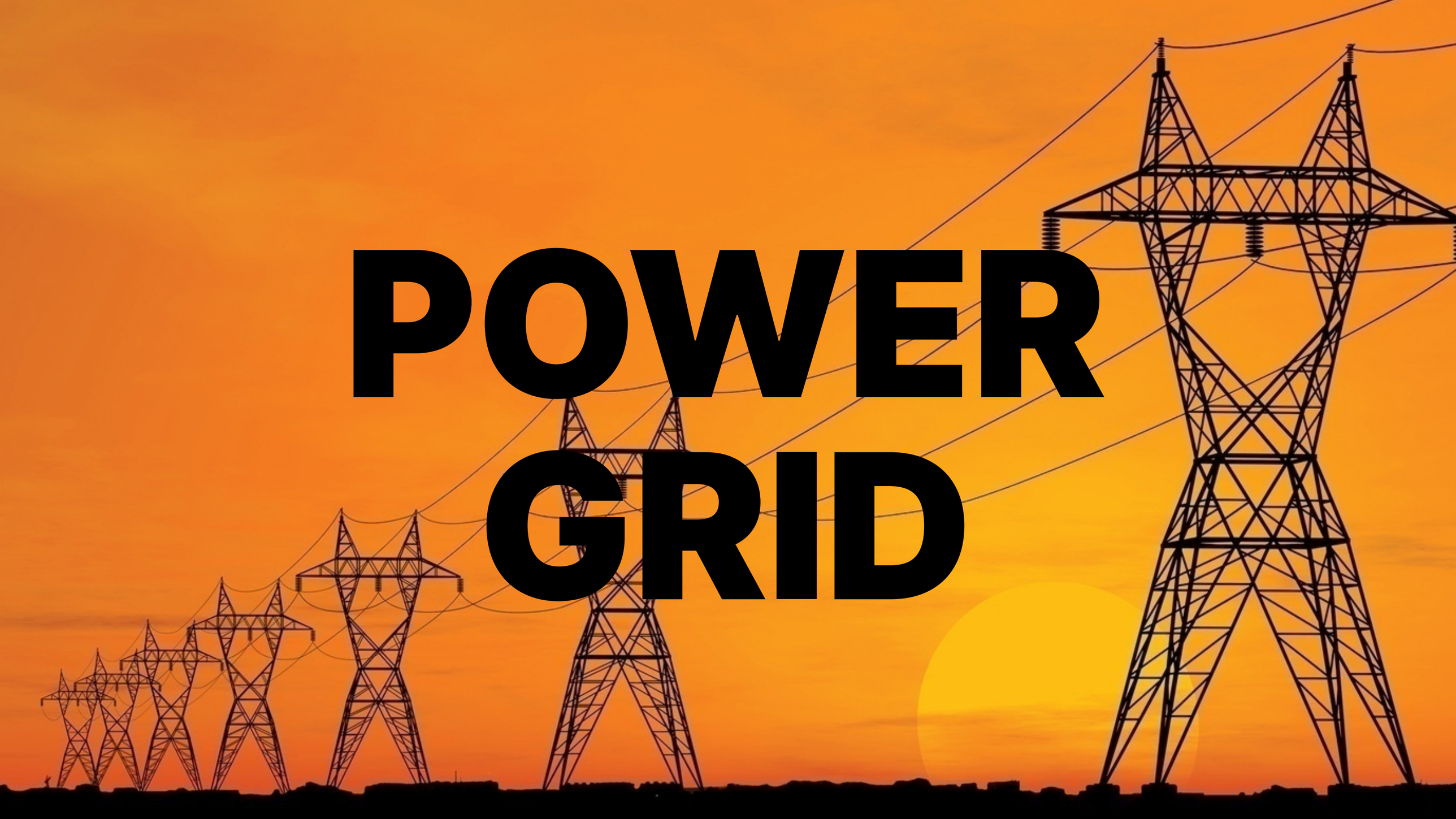
SUPPLY CHAINS

Resource extraction

Global shipping & transport

International trade

Just-in-time production

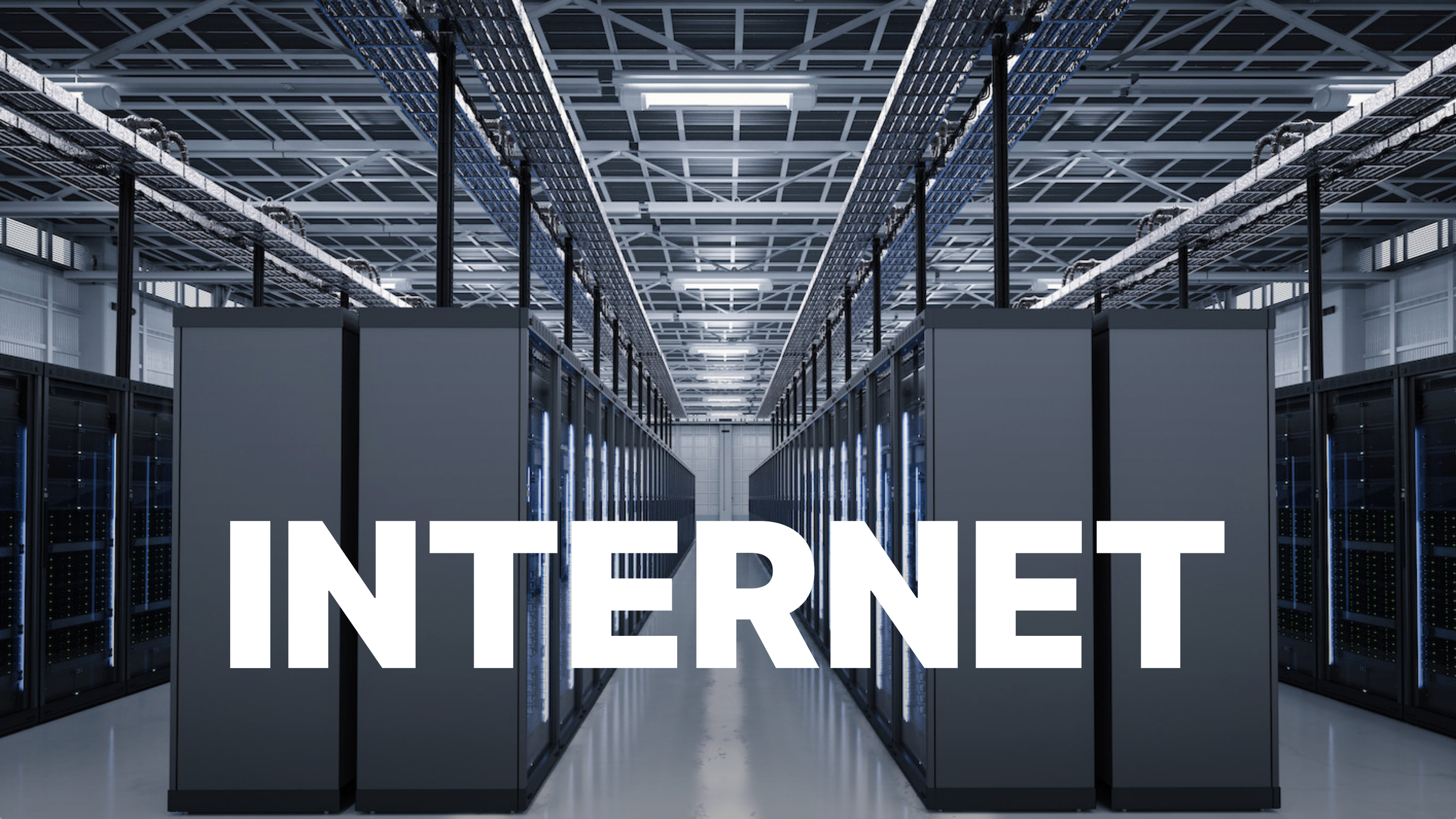

POWER
GRID

Air conditioning

Public services & transport

Computers

Network infrastructure

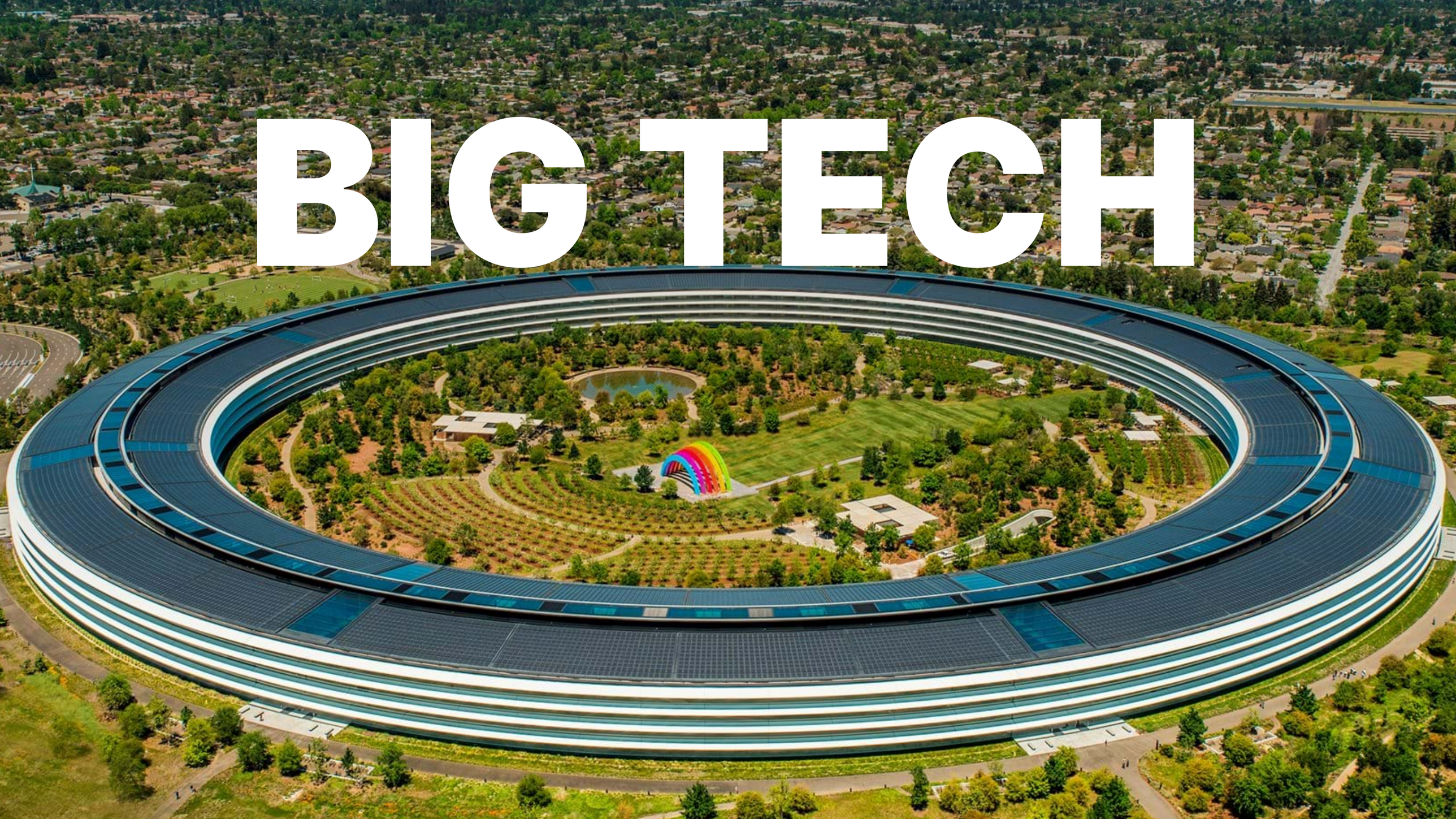

INTERNET

Infrastructure & industry

Maps & GPS

Information resources

Communication


BIG TECH

OS udpates & app stores

Cloud infrastructure & services

SaaS (Notion, Figma, etc.)

Github!

No new hardware

Limited power

Limited connectivity

No cloud services

Global south as a preview

Heat waves, droughts, floods, etc.

Wars and economic sanctions

Sporadic power and internet access

Limited access to consumer goods

What will the world look like?*

Localized

Disconnected

Resource-constrained

Self-reliant

* If we’re lucky. Nuclear war is also a possibility...

To be extra clear:
Fuck nazis

Small-scale, localized, resilient communities

How to do computing will not be our most pressing problem.

But it may be part of some important solutions.

Information storage & management

Local networking & communication

(Re-)programming machinery

Sensors & automation

Free software may have a critical role to play in this future

But maybe we need to rethink a few things?

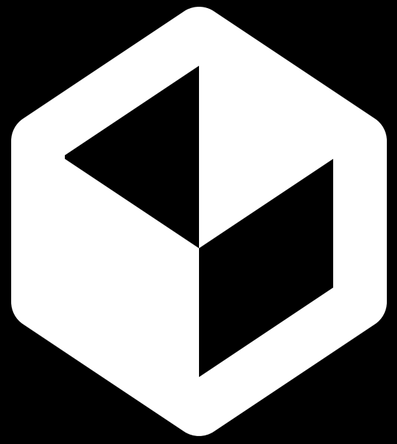
Current Assumptions

Fast, always-available internet

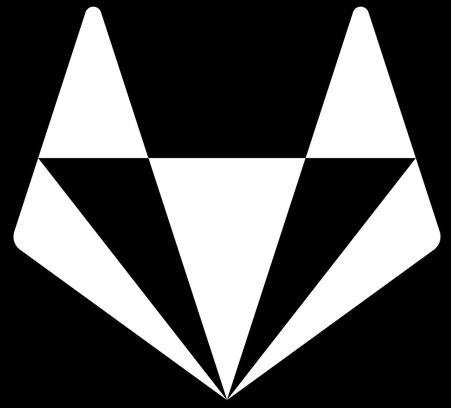
New hardware every few years

Everything will get faster over time

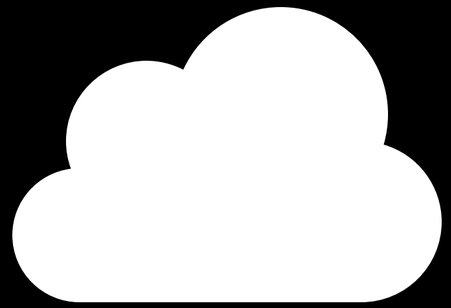
Use all the resources available



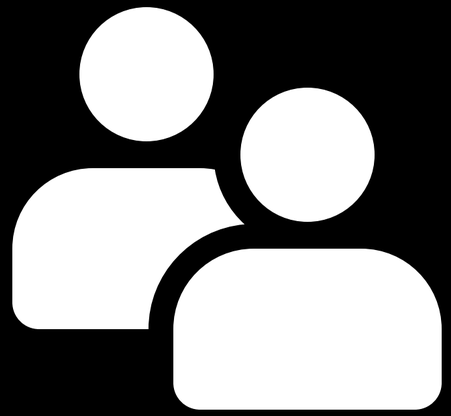
Dependencies & package managers?



Web-based tooling?

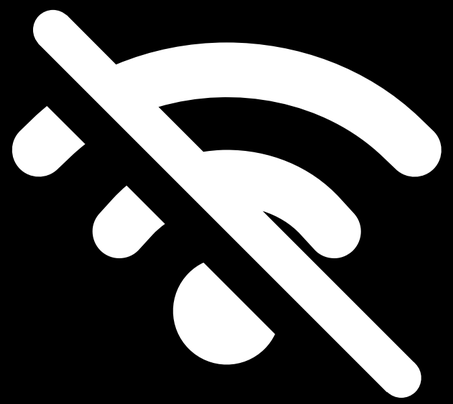


Cloud storage & streaming?



Gobal developer community without the internet?

What can we work on today to prepare for this future?



Local First

Local First

Internet is nice to have, never required

Sync and collaboration using CRDTs

Local caching for everything

USB app installation & updates

Local-First Software:
You Own Your Data, in spite of the Cloud

Martin Kleppmann
Department of Computer Science and Technology
University of Cambridge
Cambridge, United Kingdom
martin.kleppmann@cl.cam.ac.uk

Adam Wiggins
Ink & Switch
Berlin, Germany
adam@inkandswitch.co

Peter van Hardenberg
Ink & Switch
San Francisco, CA, USA
pvh@inkandswitch.com

Mark McGranagha
Ink & Switch
Seattle, WA, USA
mark@inkandswitch.co

Abstract

Cloud apps like Google Docs and Trello are popular because they enable real-time collaboration with colleagues, and they make it easy for us to access our work from all of our devices. However, by centralizing data storage on servers, cloud apps also take away ownership and agency from users. If a service shuts down, the software stops functioning, and data created with that software is lost.

In this article we propose *local-first software*, a set of principles for software that enables both collaboration *and* ownership for users. Local-first ideals include the ability to work offline and collaborate across multiple devices, while also improving the security, privacy, long-term preservation, and user control of data.

We survey existing approaches to data storage and sharing, ranging from email attachments to web apps to Firebase-backed mobile apps, and we examine the trade-offs of each. We look at Conflict-free Replicated Data Types (CRDTs): data structures that are multi-user from the ground up while also being fundamentally local and private. CRDTs have the potential to be a foundational technology for realizing local-first software.

We share some of our findings from developing local-first software prototypes at the Ink & Switch research lab over the course of several years. These experiments test the viability of CRDTs in practice, and explore the user interface challenges for this new data model. La
next steps for moving towards local-
searchers, for app developers, and a st
entrepreneurs.

CCS Concepts • **Human-centered laborative content creation**; *Ubiqui puting systems and tools*; • **Compute tion** → *Peer-to-peer architectures*; • **So neering** → *Peer-to-peer architectures*; *for web applications.*

Keywords collaboration software, m ownership, CRDTs, peer-to-peer comm

ACM Reference Format:
Martin Kleppmann, Adam Wiggins, Peter va McGranaghan. 2019. Local-First Software: spite of the Cloud. In *Proceedings of the ternational Symposium on New Ideas, New tions on Programming and Software (Onw 2019, Athens, Greece.* ACM, New York, NY //doi.org/10.1145/3359591.3359737

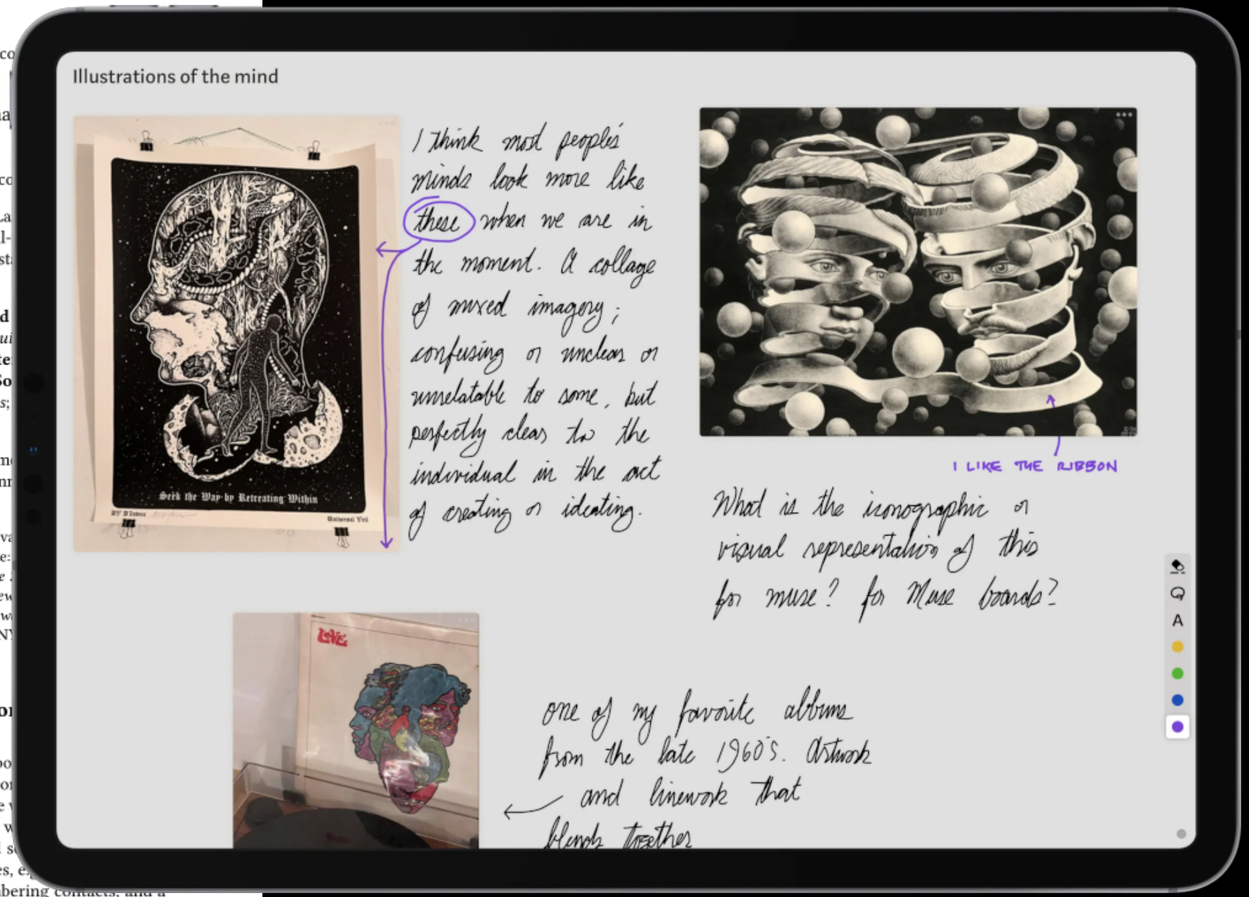
1 Motivation: Collaboratio Ownership

It's amazing how easily we can collabo We use Google Docs to collaborate o sheets and presentations; in Figma we interface designs; we communicate w Slack; we track tasks in Trello; and s these and many other online services, e planning projects or events, remembering whole raft of business uses.

We will call these services "cloud apps," but you could just as well call them "Software as a Service" (SaaS) or "web-based apps." What they have in common is that we typically access them through a web browser or through mobile apps, and that they store their data on a server.

Onward! '19, October 23–24, 2019, Athens, Greece

ACM ISBN 978-1-4503-6995-4/19/10…$15.00
https://doi.org/10.1145/3359591.3359737



USB Drive

SanDisk 32GB

9 apps, 3 system updates
8.6 GB free

Offline Updates Available
Install updates for 4 of your apps, and a system update from this drive
5

Copy Apps or Updates to Drive
Add installed software to the drive to install on other computers

Apps

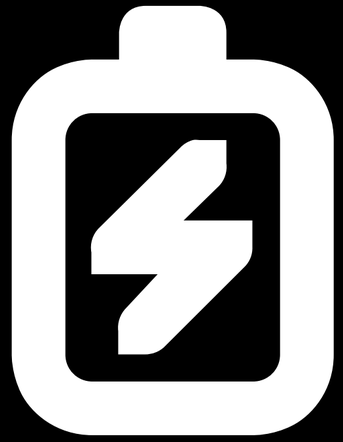
Install these apps offline from the drive

Tangram
Web apps, on your desktop

Shortwave
Listen to internet radio

Tootle

Apostrophe



Resource Efficiency

Resource Efficiency

Reduce CPU/memory/power use

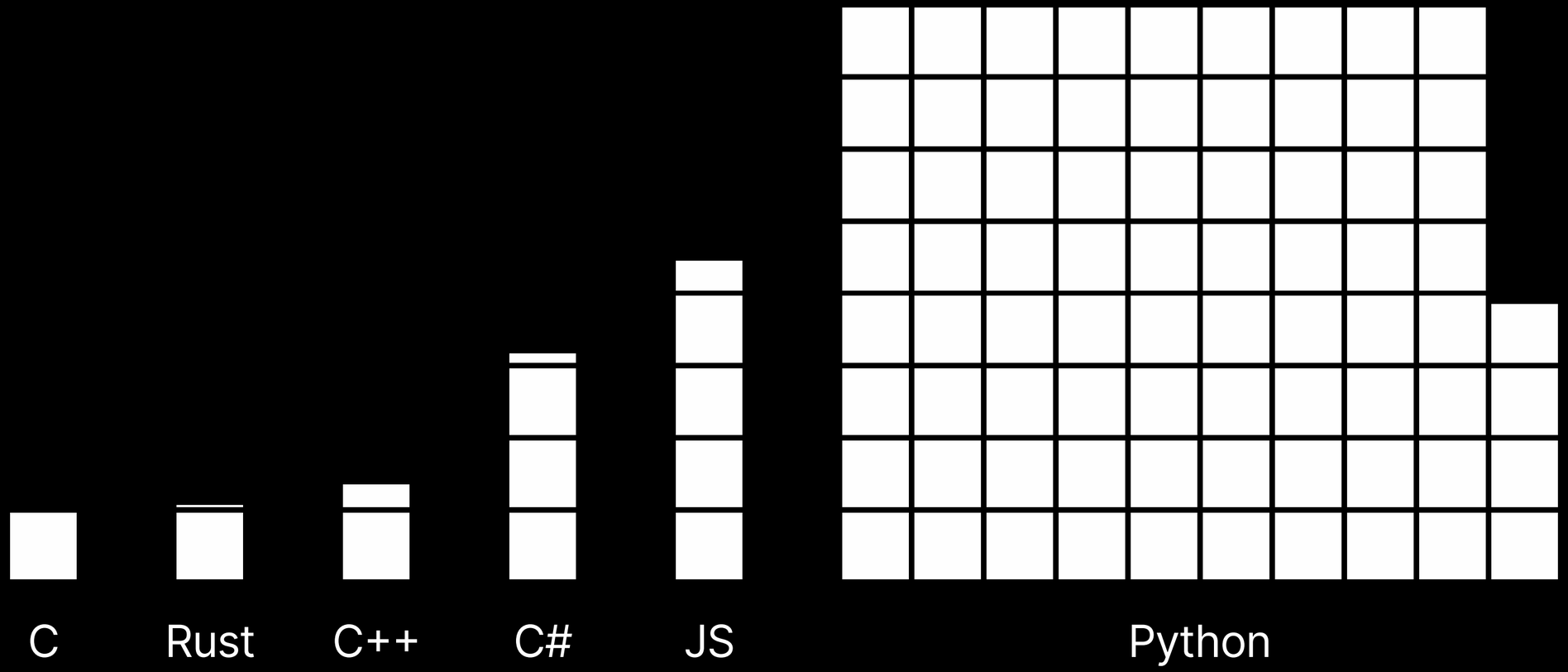
Improve tooling to make this easier

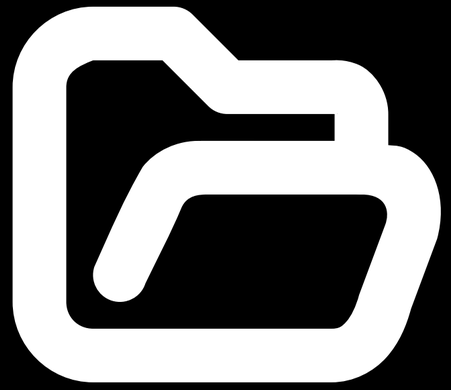
Avoid web tech, use GTK

Schedule tasks when there is power

Programming Language Energy Use

Source: Pereira et al., *Energy Efficiency across Programming Languages*, 2017





Data Resilience

Data Resilience

Keep local copies of everything

The file system is good, actually

Simple, standardized formats

Human-readable text fallbacks

TextBundle Specification

The purpose of the TextBundle file format is to simplify the exchange of various plain text files together with additional images between sandboxed applications. By using a single package file, applications do not need to acquire additional sandbox extensions to access files referenced by a plain text document. It can be used as container for various plain text formats, like Markdown or Fountain.

File Type

There are two variants of the TextBundle format: The *package format* and the *compressed format*.

Package Format

The package format is designed to simplify exchange between applications: For example, it can be used to pass a Markdown file together with its assets from an editor to an export tool.

The package format uses the path extension `.textbundle`. The UTI of a TextBundle package is `org.textbundle.package` which inherits from the generic package UTI `com.apple.package`.

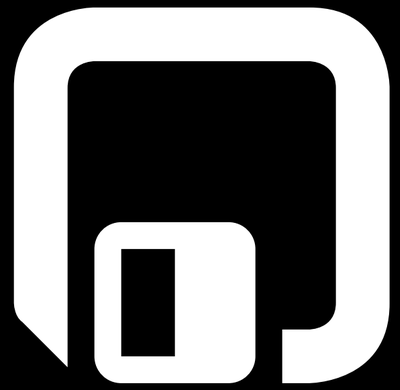
Compressed Format

Alternatively, TextBundles can be stored inside a compressed Zip container. This is especially useful for exchanging files between users.

The compressed format uses the path extension `.textpack`. The UTI of compressed TextBundles is `org.textbundle.compressed` which inherits from the ZIP format UTI `com.pkware.zip-archive`.

Deriving Own Formats

Generally, you are free to derive your own format from TextBundle. In this case, your subformat just needs to use a different path extension and inherits its UTI either from `org.textbundle.package`



Keep Old Hardware Running

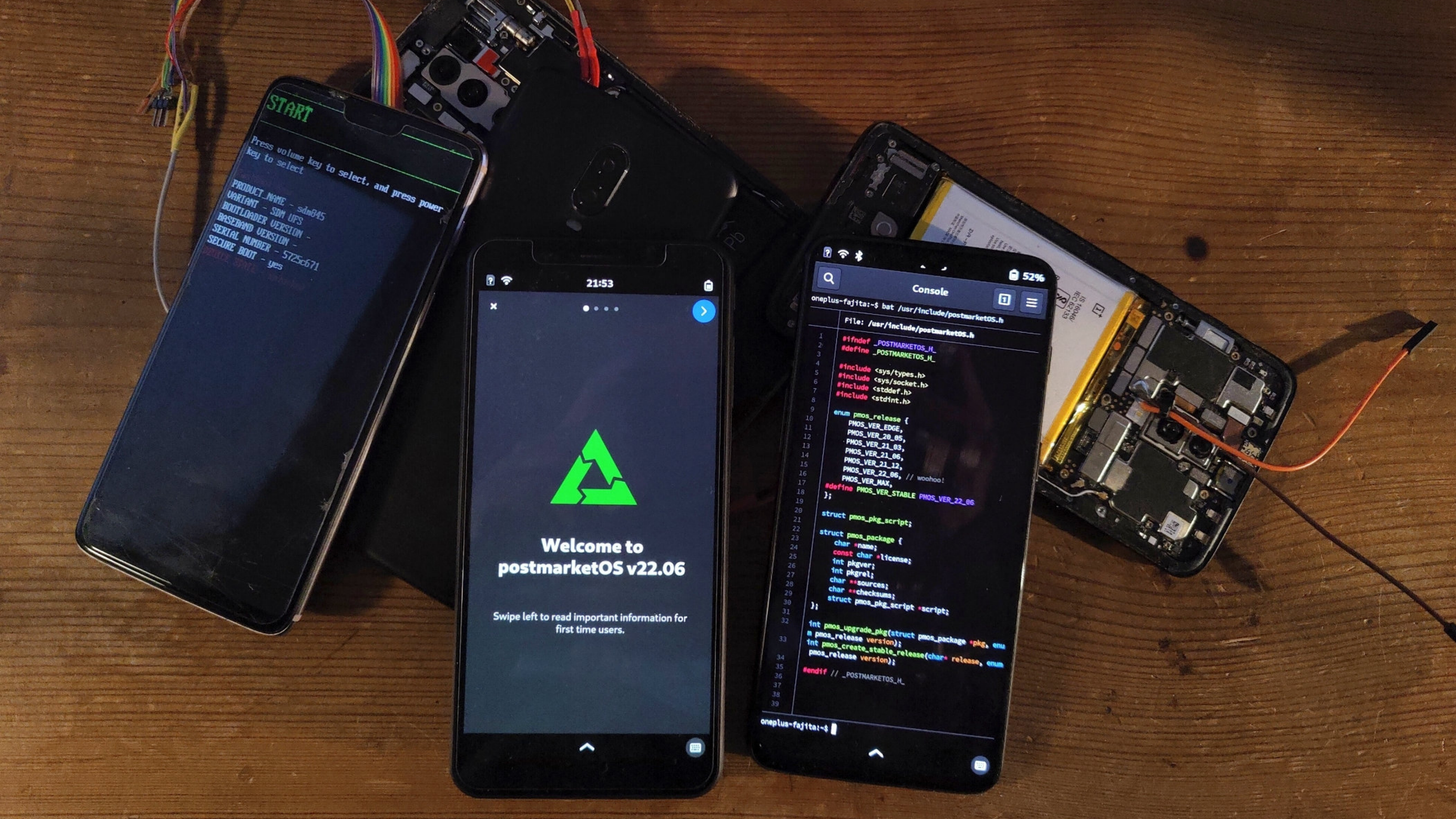
Keep Old Hardware Running

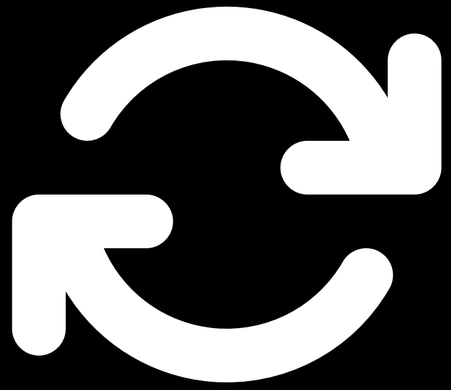
Reducing resource use helps here, too

Support as many form factors as possible

Reverse engineer common hardware

Hack bootloaders on locked devices


START
Press volume key to select, and press power key to select
PRODUCT_NAME - sdm845
VARIANT - SDM UFS
BOOTLOADER VERSION -
BASEBAND VERSION -
SERIAL NUMBER - 5725c671
SECURE BOOT - yes
21:53
Welcome to postmarketOS v22.06
Swipe left to read important information for first time users.
52%
Console
oneplus-fajita:~$ bat /usr/include/postmarketOS.h
File: /usr/include/postmarketOS.h
#ifndef _POSTMARKETOS_H_
#define _POSTMARKETOS_H_
#include <sys/types.h>
#include <sys/socket.h>
#include <stddef.h>
#include <stdint.h>
enum pmos_release {
PMOS_VER_EDGE,
PMOS_VER_20_05,
PMOS_VER_21_03,
PMOS_VER_21_06,
PMOS_VER_21_12,
PMOS_VER_22_06, // woohoo!
PMOS_VER_MAX,
#define PMOS_VER_STABLE PMOS_VER_22_06
};
struct pmos_pkg_script;
struct pmos_package {
char *name;
const char *license;
int pkgver;
int pkgrel;
char **sources;
char **checksums;
struct pmos_pkg_script *script;
};
int pmos_upgrade_pkg(struct pmos_package *pkg, enu
m pmos_release version);
int pmos_create_stable_release(char* release, enum
pmos_release version);
#endif // _POSTMARKETOS_H_
oneplus-fajita:~$



Build for Repair

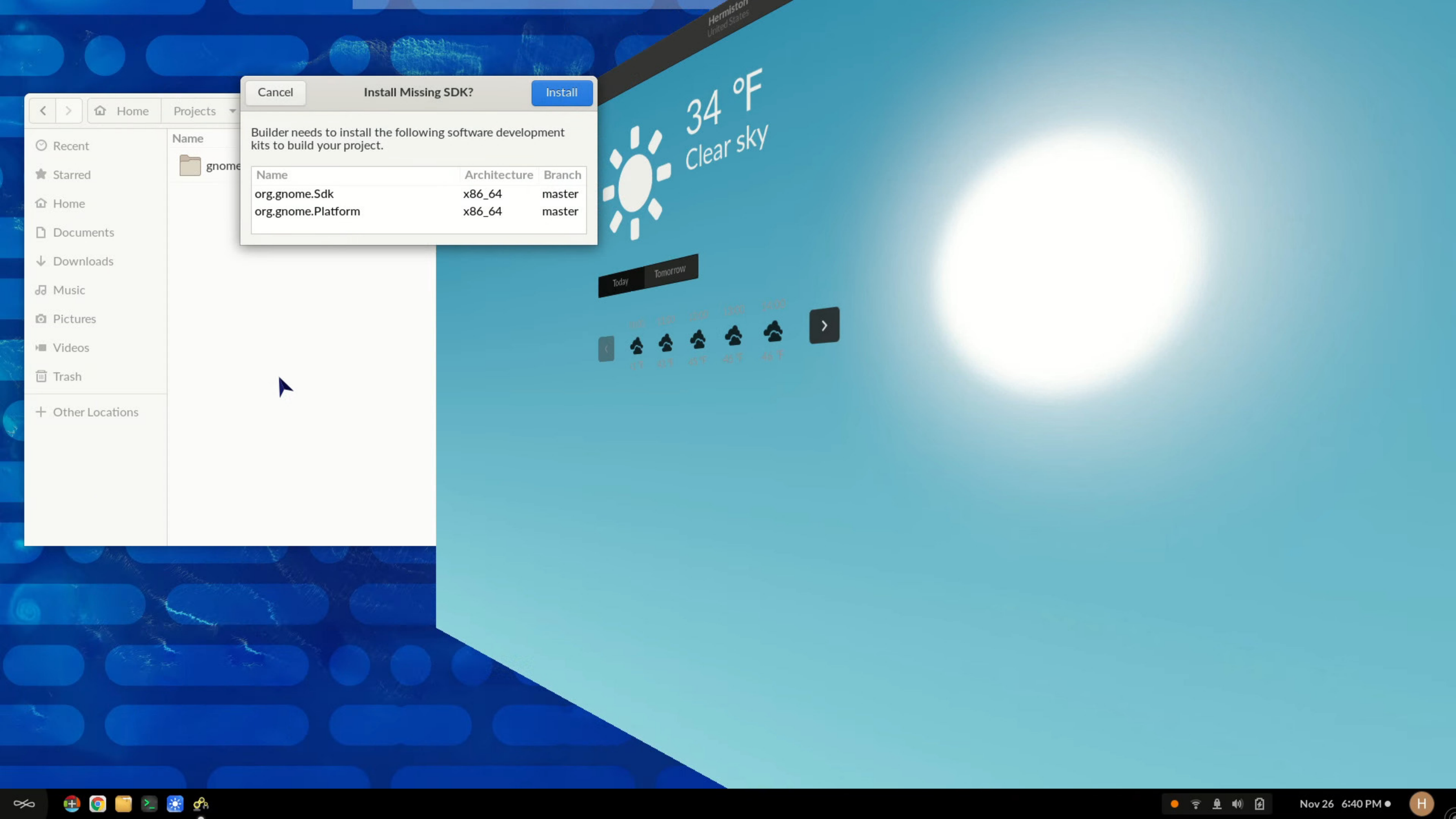
Build for Repair

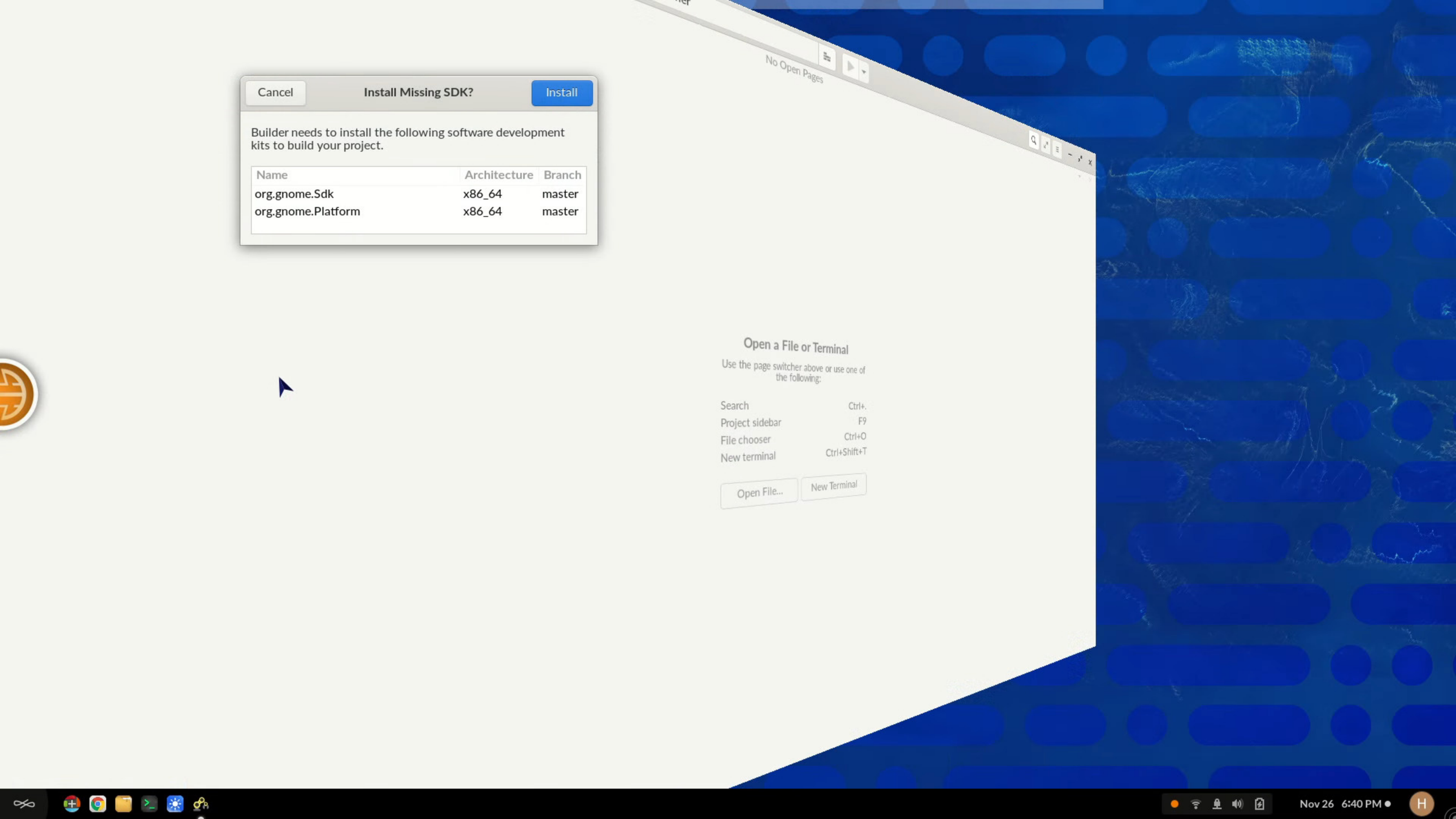
Use well-understood and accessible tech

Make everything easy to build offline

Great offline documentation and tutorials

Ship source and docs with everything?


Cancel
Install Missing SDK?
Install
Builder needs to install the following software development kits to build your project.
Name
Architecture
Branch
org.gnome.Sdk
x86_64
master
org.gnome.Platform
x86_64
master
Home
Projects
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
Name
gnome
34 °F
Clear sky
Today
Tomorrow
Nov 26 6:40 PM


Cancel
Install Missing SDK?
Install
Builder needs to install the following software development kits to build your project.
Name
Architecture
Branch
org.gnome.Sdk
x86_64
master
org.gnome.Platform
x86_64
master
No Open Pages
Open a File or Terminal
Use the page switcher above or use one of the following:
Search
Ctrl+.
Project sidebar
F9
File chooser
Ctrl+O
New terminal
Ctrl+Shift+T
Open File...
New Terminal
Nov 26
6:40 PM

Efficient vs. Repairable

What else?

People to learn from?

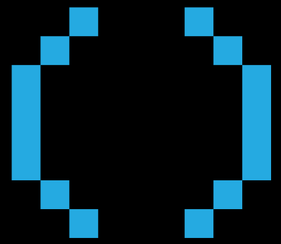
Permacomputing

100Rabbits

Right to Repair

Solarpunk

How could we fund work towards this?



OPEN
TECHNOLOGY
FUND

Prototype
Fund

SOVEREIGN TECH FUND

Takeaways

Business as usual isn’t an option

Civil resistance is our best hope

Making our tech resilient is good regardless

See you in action o/

SEP 17-20

FALL REBELLION IN BERLIN