

Preparing Next Mobile Explosion : a glimpse of multi-device operating system

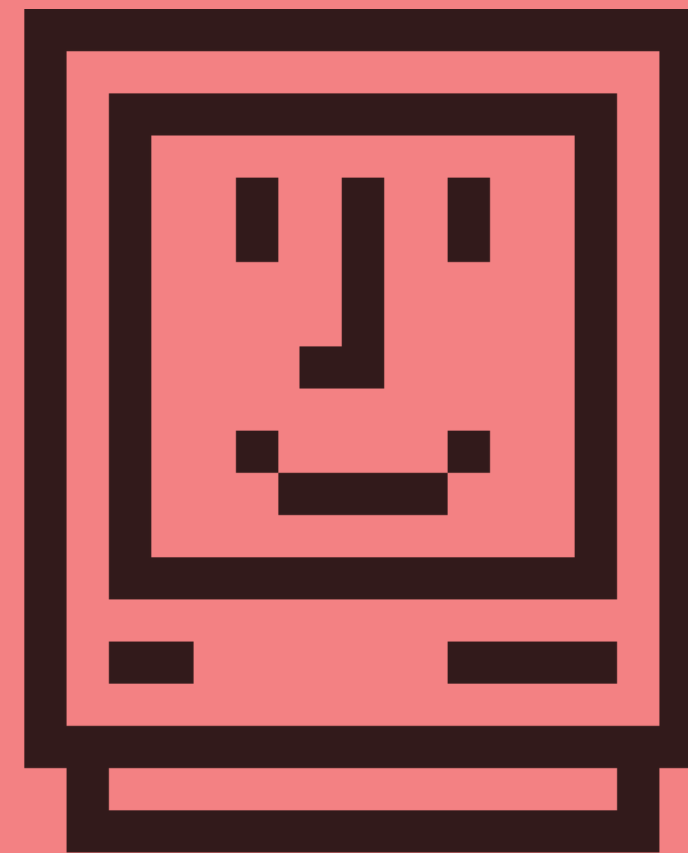


TEAM #7

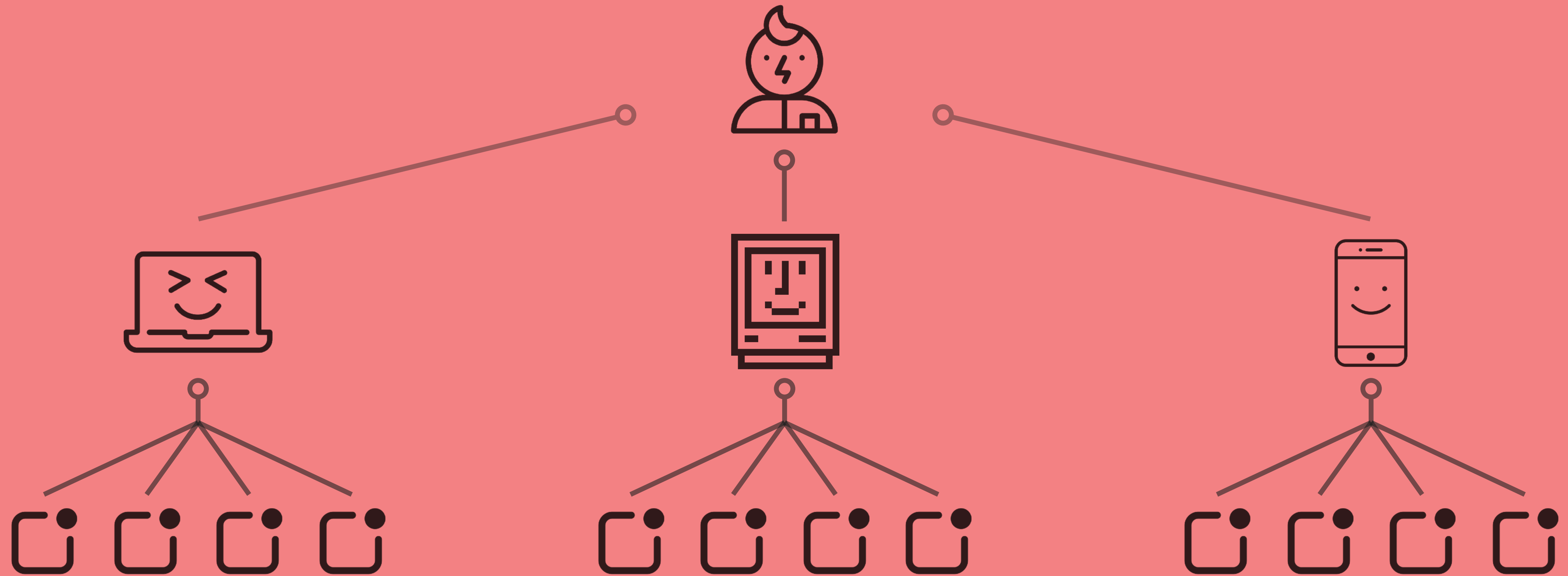
Youngjae Chang youngjae.chang@kaist.ac.kr

Tae Gyeong Lee danny003@kaist.ac.kr

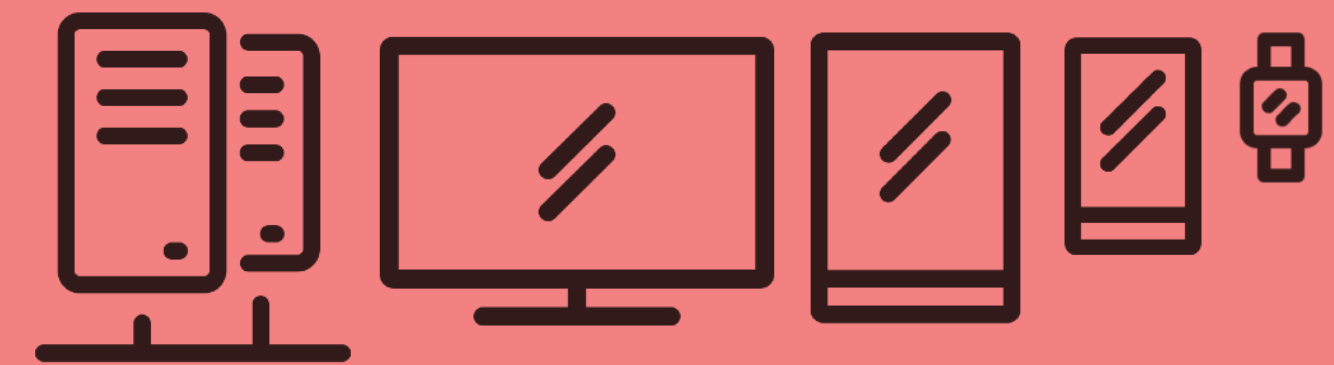
Early 2000's



Met a computer



**We have ...
Multiple devices !**

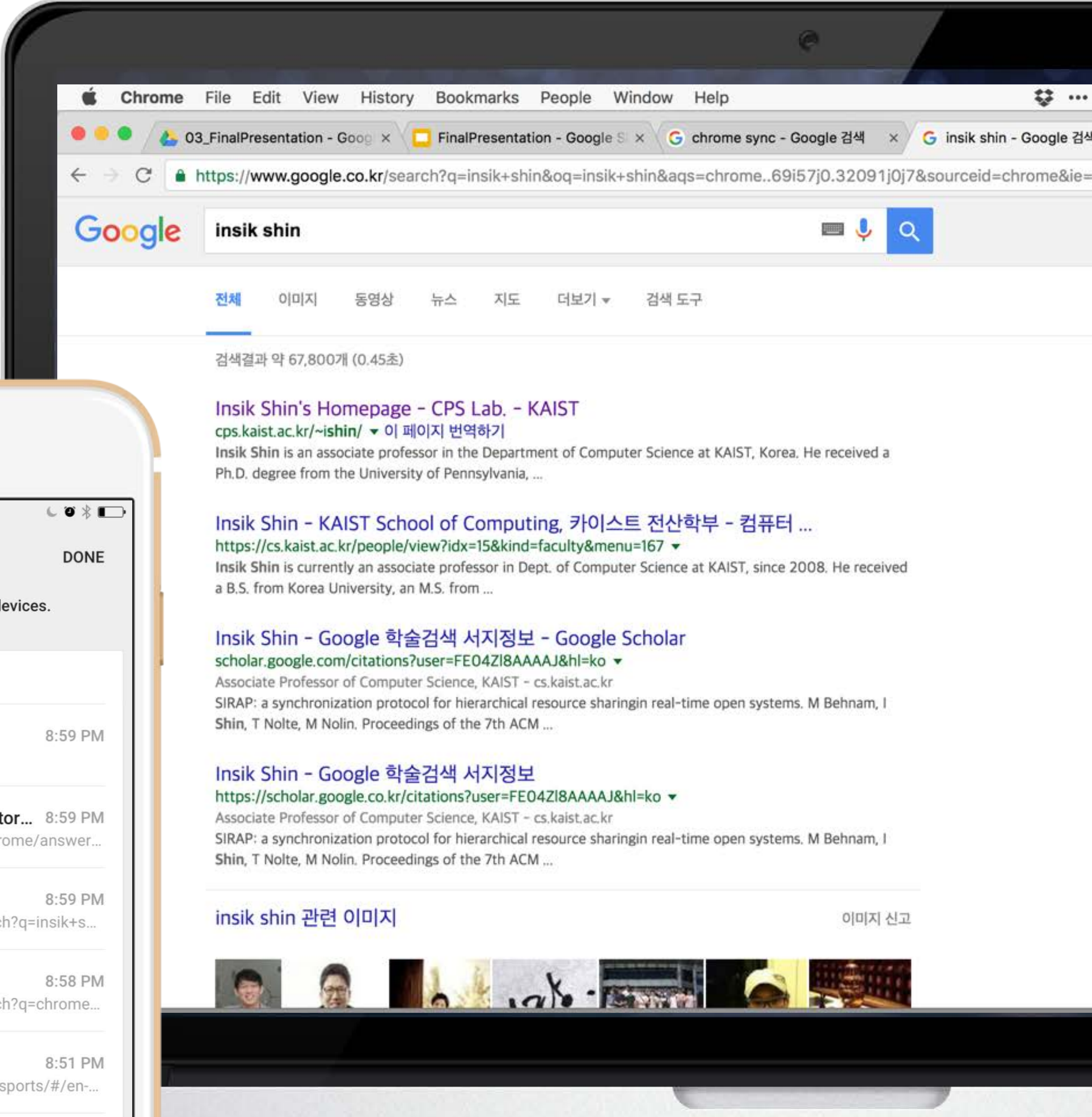
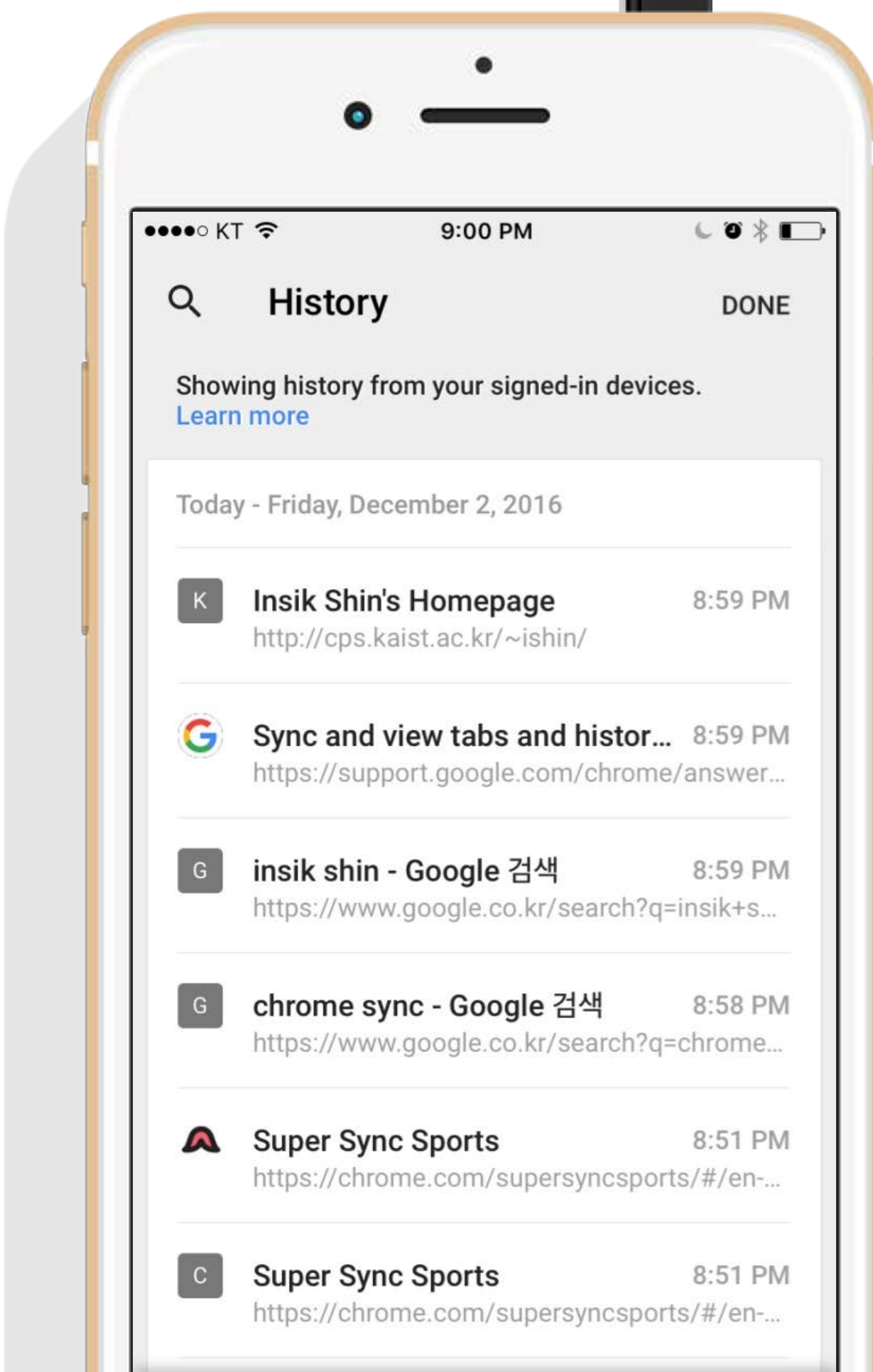


How's your experience?

Platform arose from examples!

Examples

I. Chrome

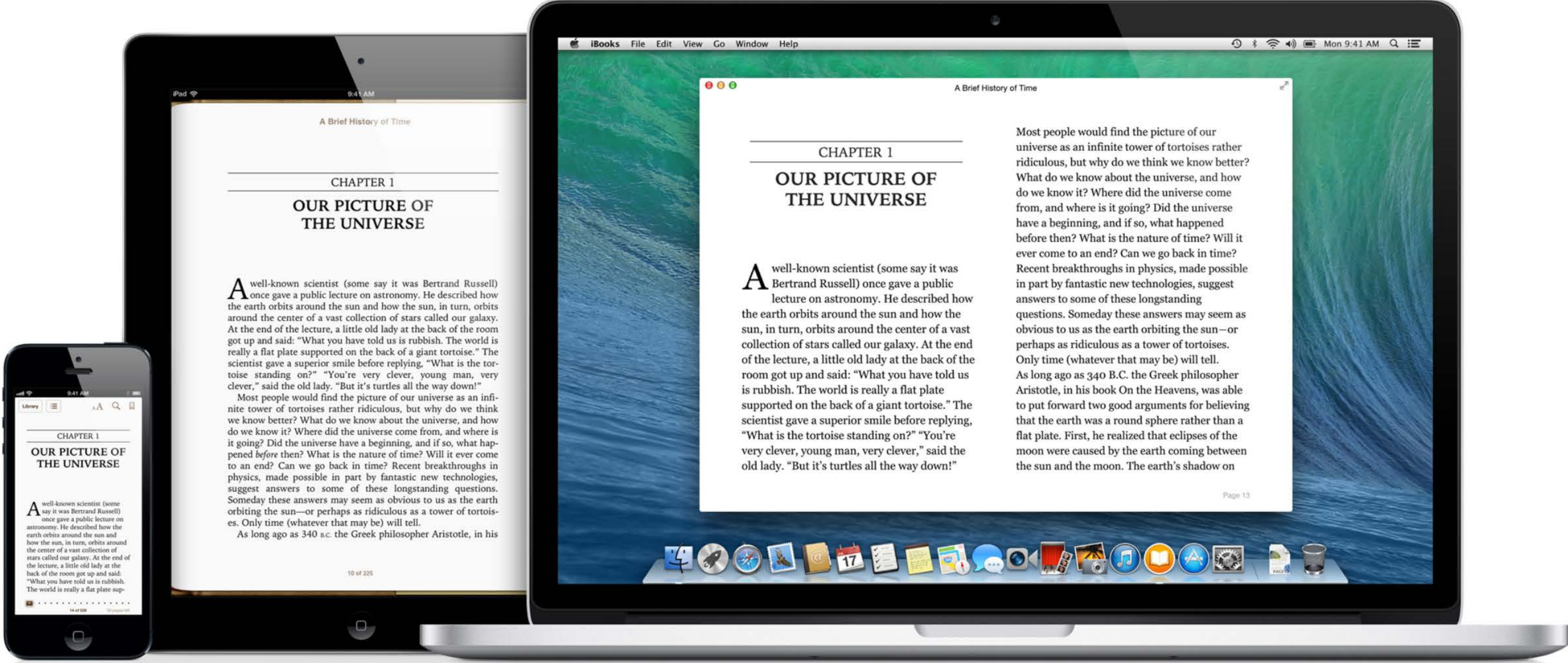


2. Dropbox

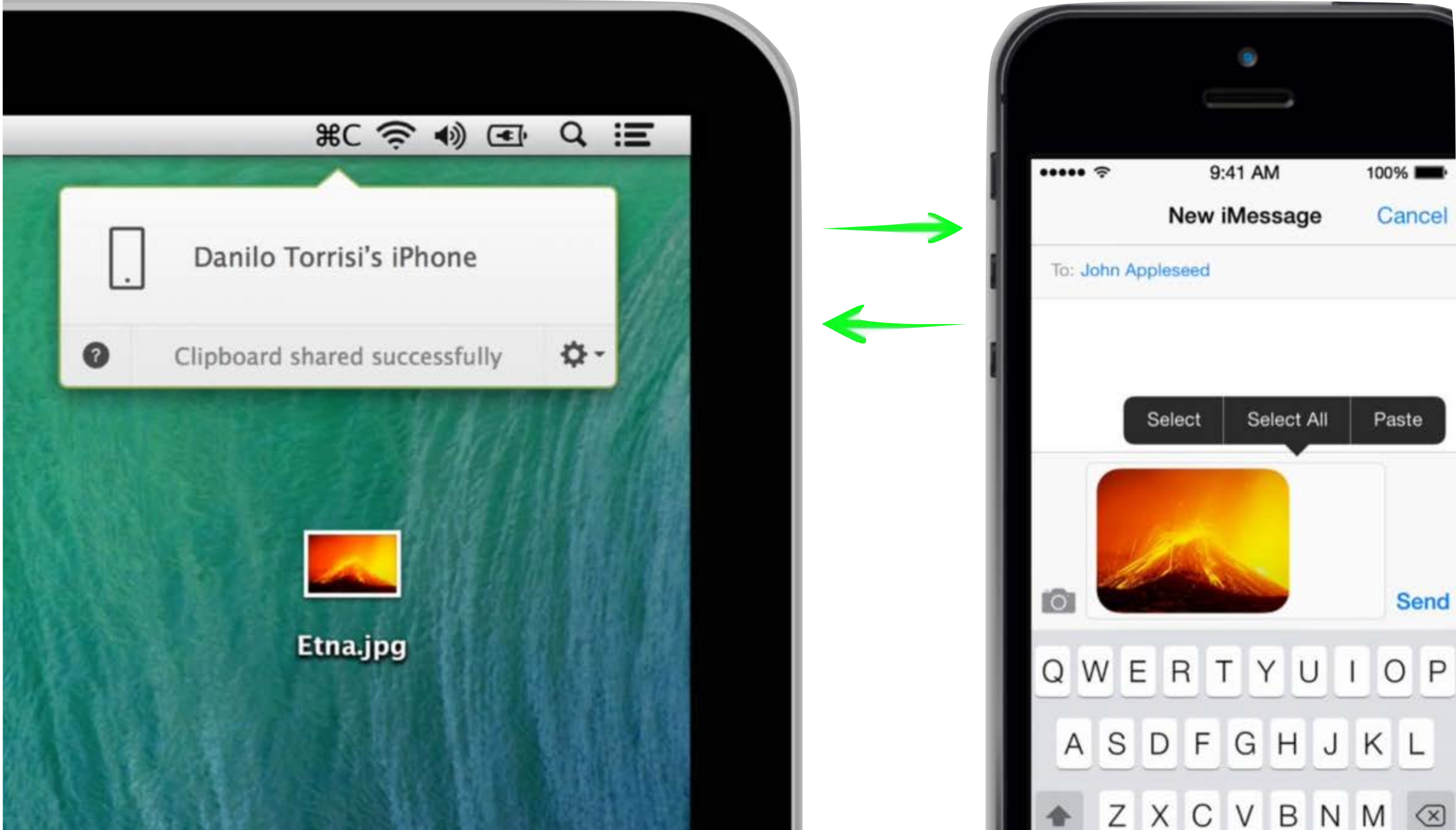


Examples

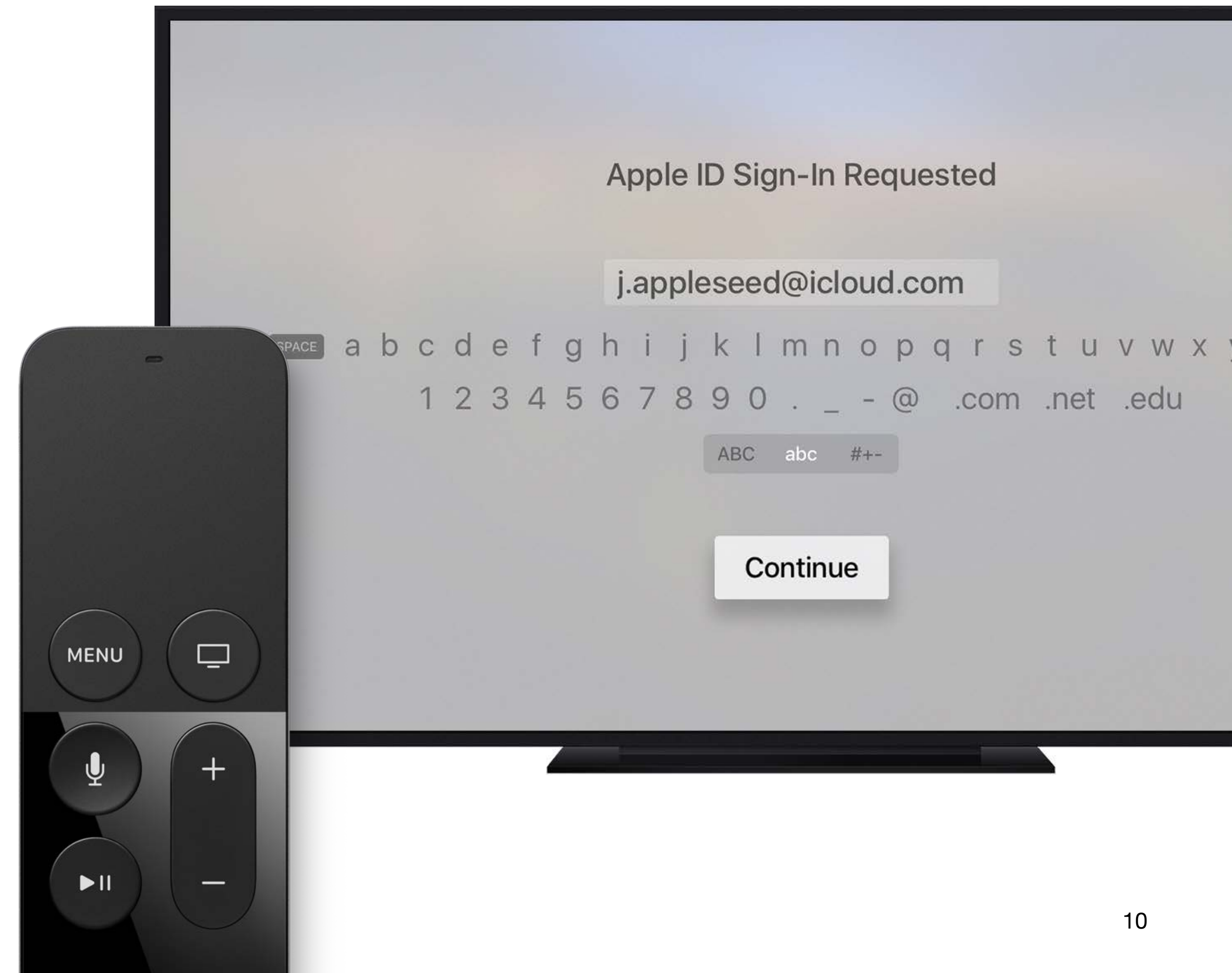
3. iBooks



4. Clipboard Sharing



5. Typing on TV



6. Handoff Calling



7. Video Encoding



8. Smartphone as a joystick!



1980:



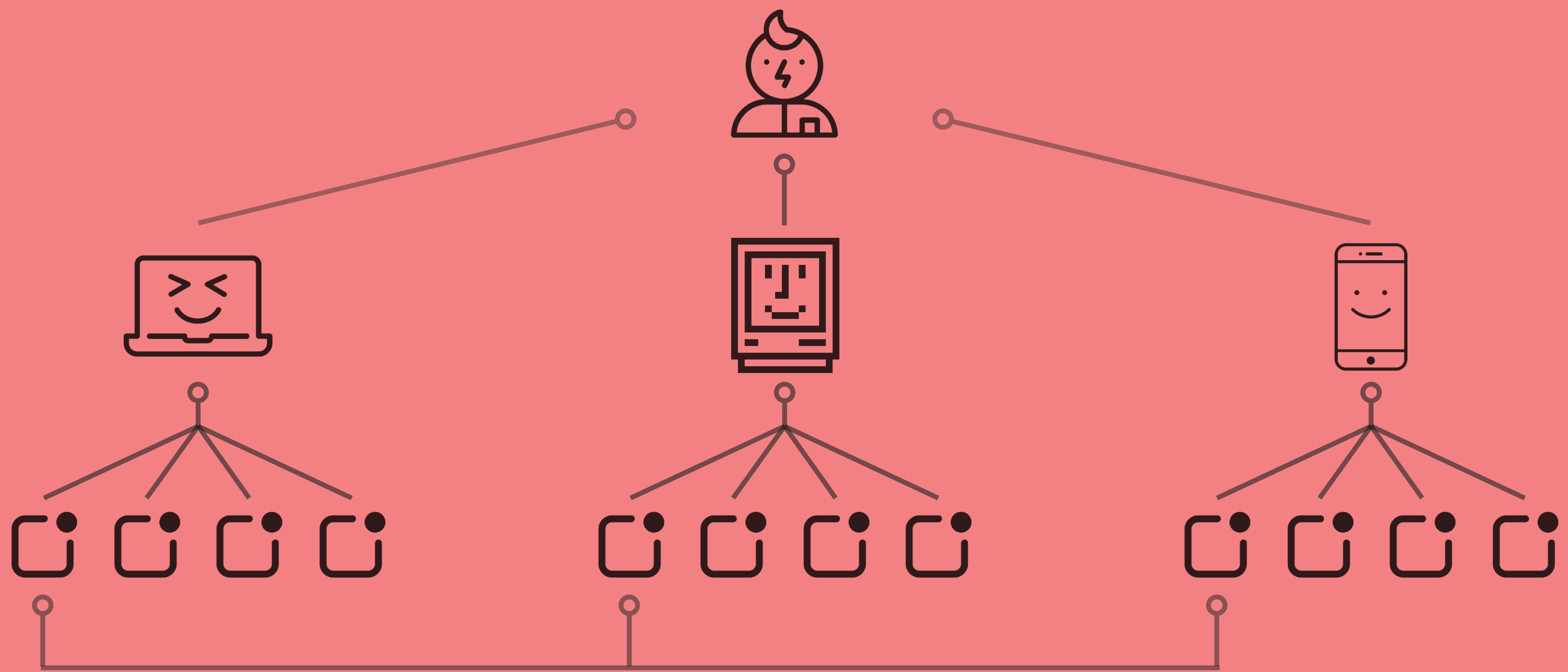
**“ A computer on every desk,
and in every home. — *Bill Gates***

“ It’s a world of multiple screens, smart displays, with tons of low-cost computing, with big sensors built into devices.

— *Sundar Pichai*

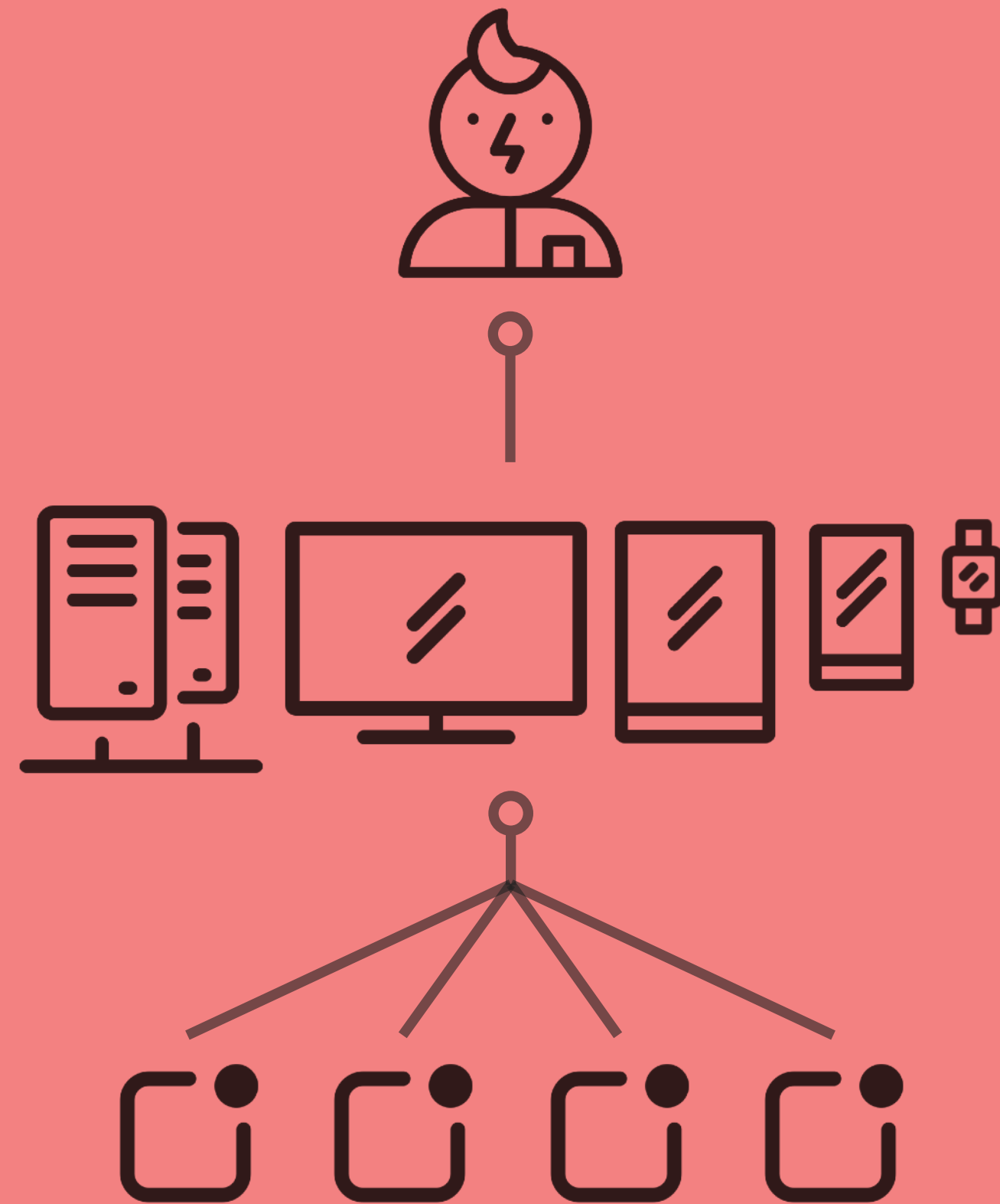
2016:

At Google, we ask how to bring together something seamless and beautiful and intuitive across all these screens.



Same app; build it 3 times?

Multi-Device Operating System



**What are the
Requirements?**

User's need

**We
use it
differently**

**Touch
Keyboard
Mouse**

User's need

**We
use it
differently**

**Touch
Keyboard
Mouse**

**We
expect
same
experience**

**Sync,
sync,
sync**

User's need

**We
use it
differently**

**Touch
Keyboard
Mouse**

**We
expect
same
experience**

**Sync,
sync,
sync**

**We
want
the most of
all my HW**

**Desktop
helping
phone**

User's need

We
use it
differently

Touch
Keyboard
Mouse

We
expect
same
experience

Sync,
sync,
sync

We
want
the most of
all my HW

Desktop
helping
phone

**We
do not
want to
worry about
privacy**

User's need

**We
use it
differently**

**Touch
Keyboard
Mouse**

**We
expect
same
experience**

**Sync,
sync,
sync**

**We
want
the most of
all my HW**

**Desktop
helping
phone**

**We
do not
want to
worry about
privacy**

Requirements

**1.
Responsive
User
Interface**

**2.
App
State
Sync**

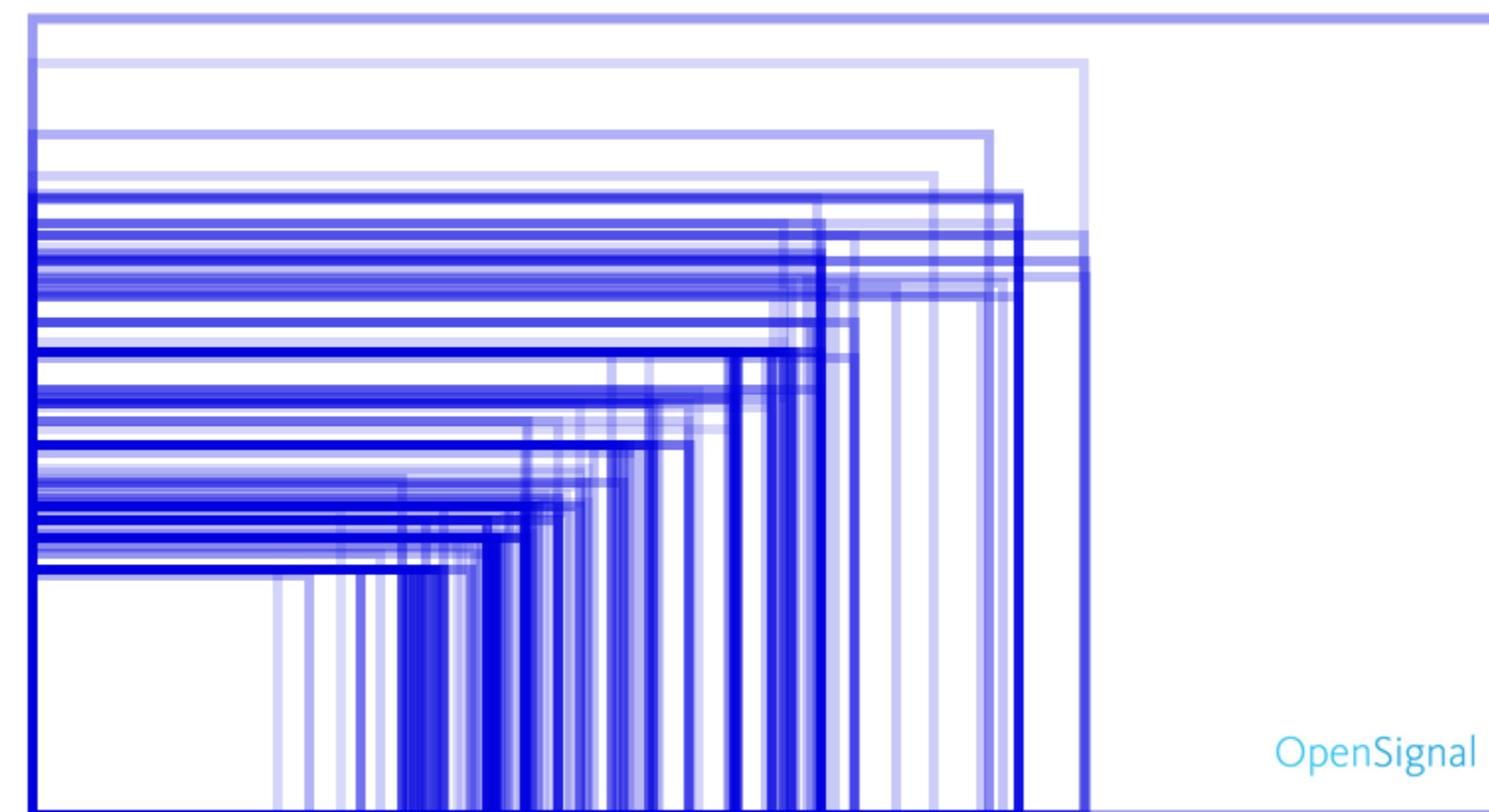
**3.
Resource
Sharing**

**4.
Privacy**

I. Responsive User Interface

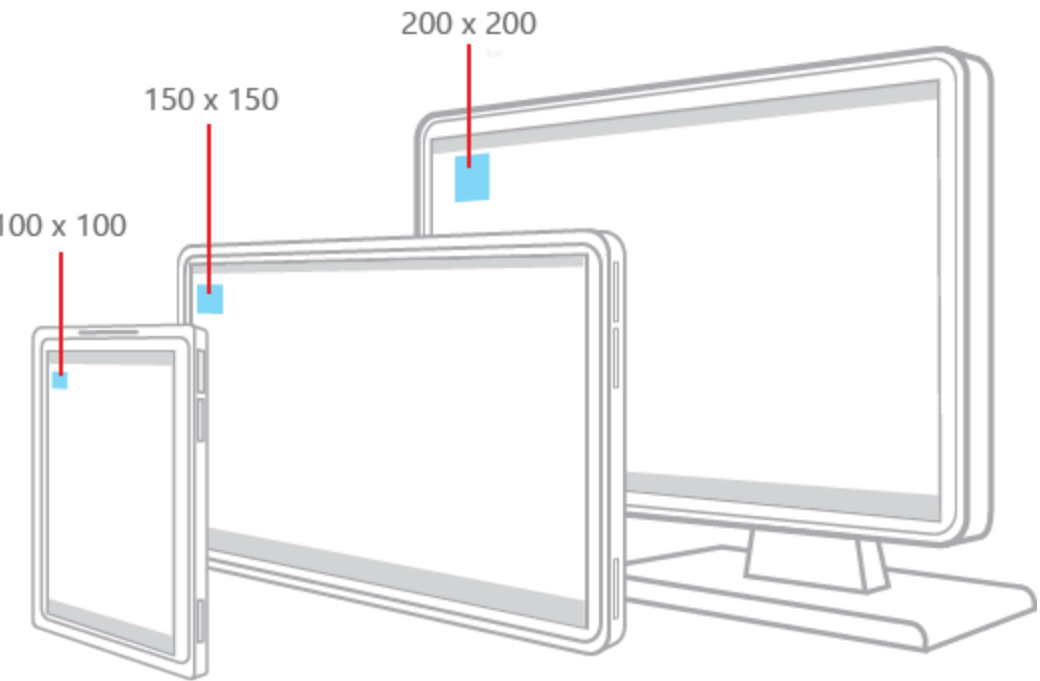
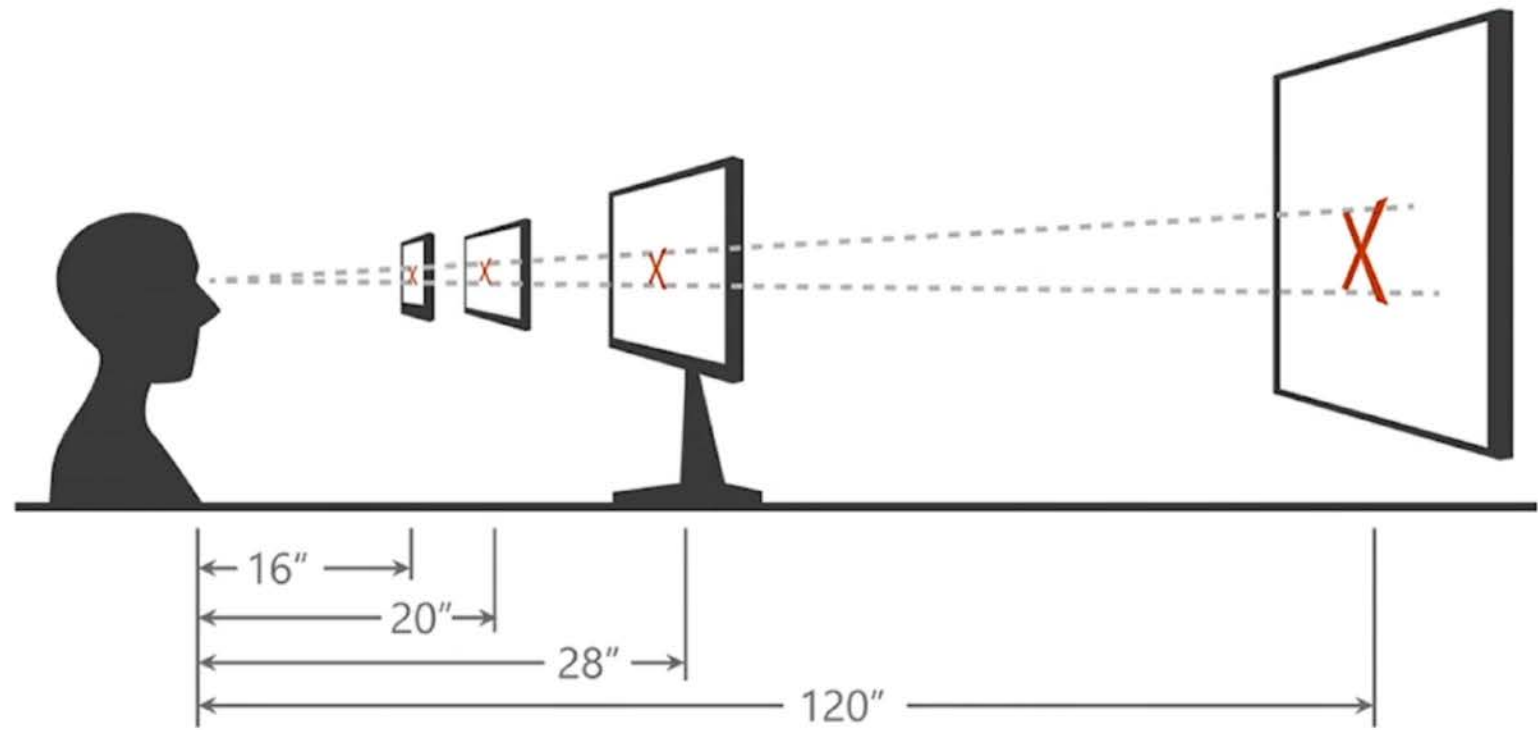
Devices require various UI

- ▶ Different Screen Size
- ▶ Different Input H/W:
Touch Screen / Mouse / Pen / Keyboard



I.
Responsive
User
Interface

Effective Pixels



2. App State Sync

Single app runs on multiple devices

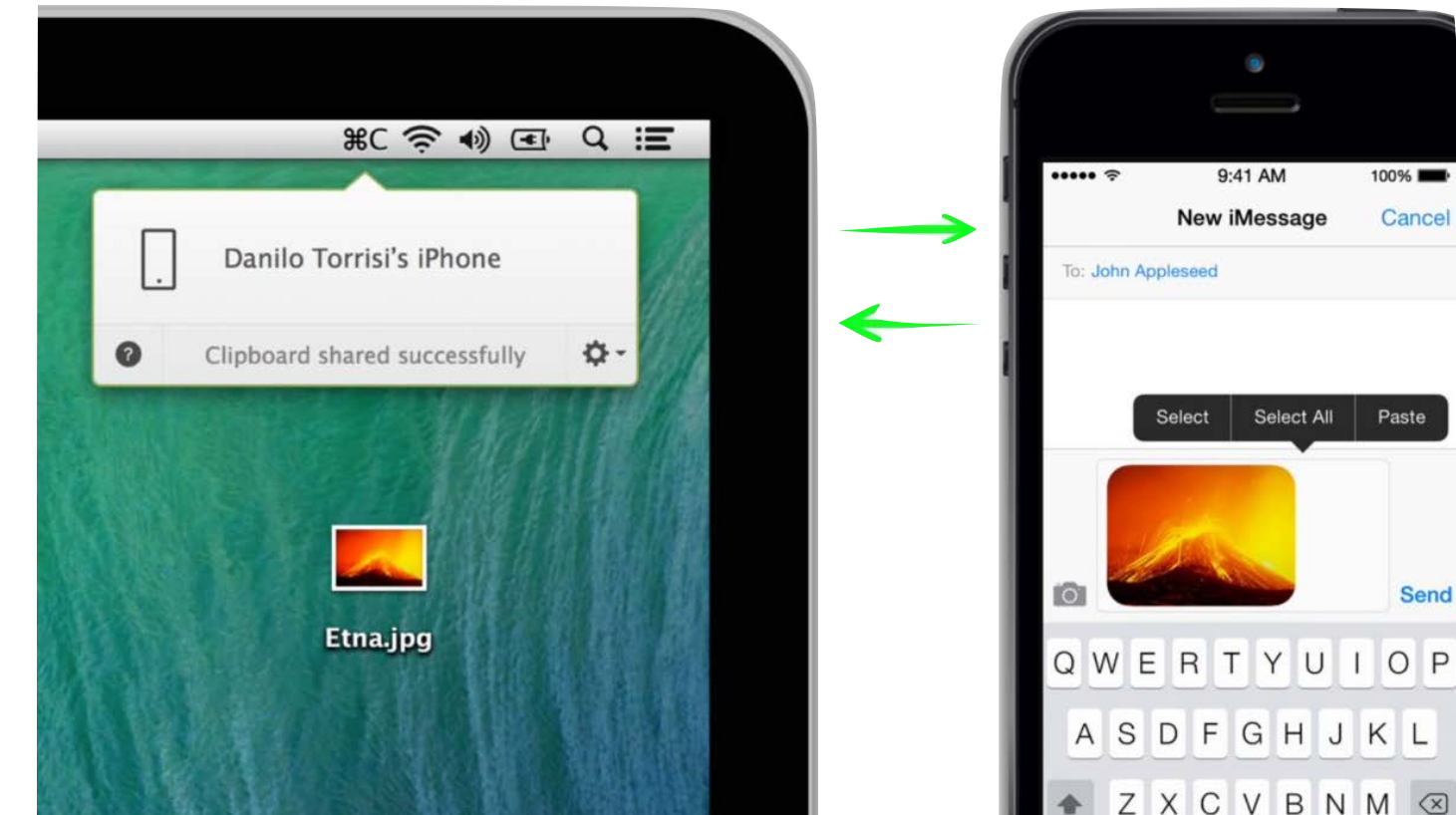
- ▶ Users try to use an app in multiple device concurrently
- ▶ App components are divided or replicated and distributed in multiple devices

Requirements

2. App State Sync



File Data



App State

3. Resource Sharing

- Devices have different capability &
Some devices are busy while others are idle**
- ▶ Mobile devices have intense physical constraints limiting its computation power
 - ▶ All devices are not created equal:
different sensors

Requirements

3. Resource Sharing



computation



sensor I/O

4. Privacy

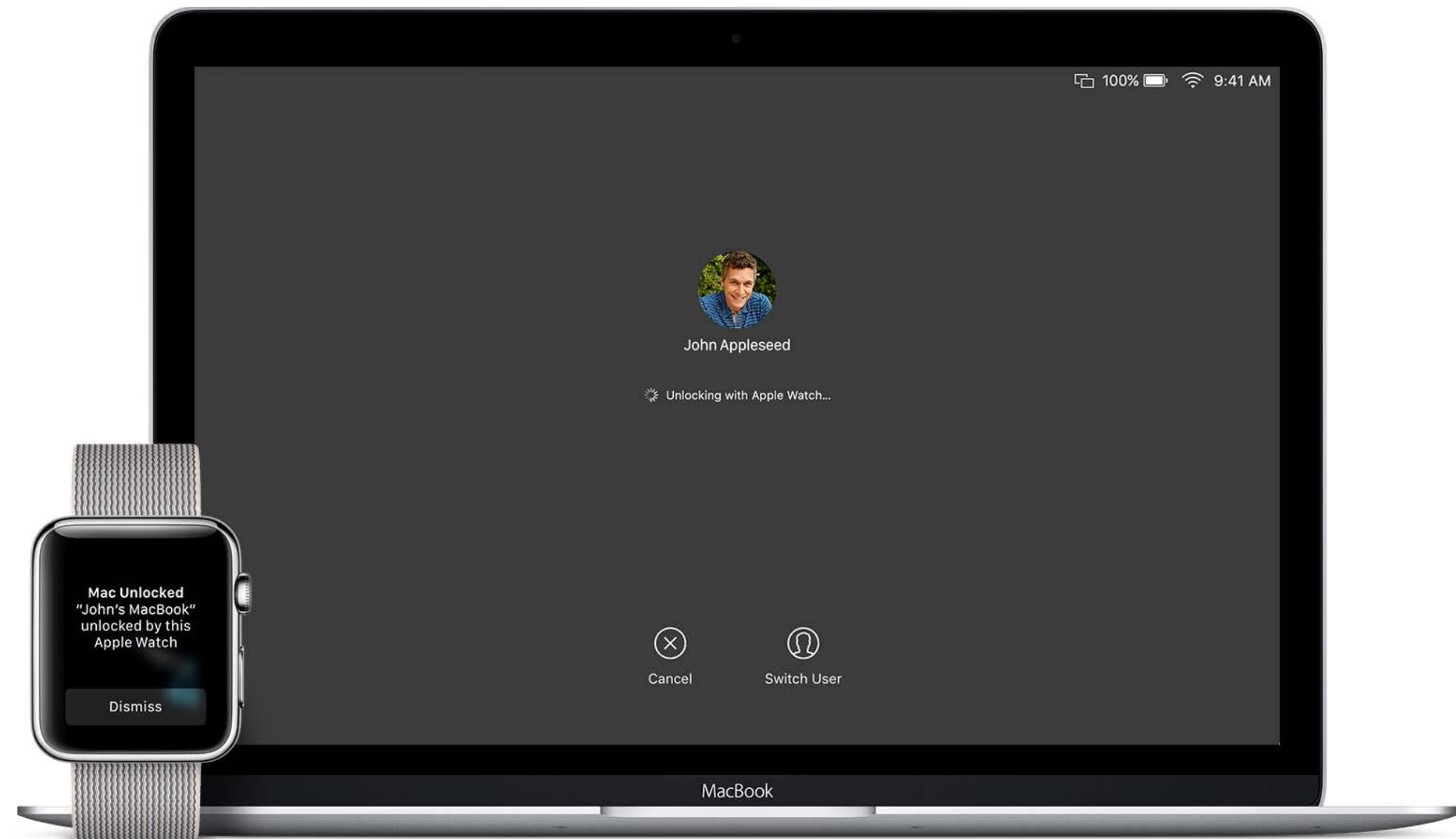
Single identity across multiple devices

- ▶ Korean people having hard time conducting online payment
- ▶ Data should have different privacy level, based on its content

Unlocking macbook when wearing apple watch

Password synchronization on Chrome


4. Privacy



unlock Mac with Apple Watch

Case Study

Case Study

	UI	App Sync	Resource Sharing	Privacy
Flux				
Rio				
Flip-Flop				

Device Heterogeneity

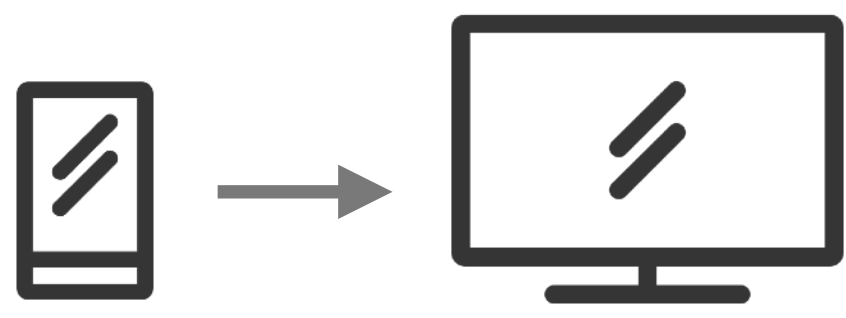
**ISA, GPU,
I/O Devices,
Screen Size
Operating System**

Network

**RTT
Connection Failure
Congestion**

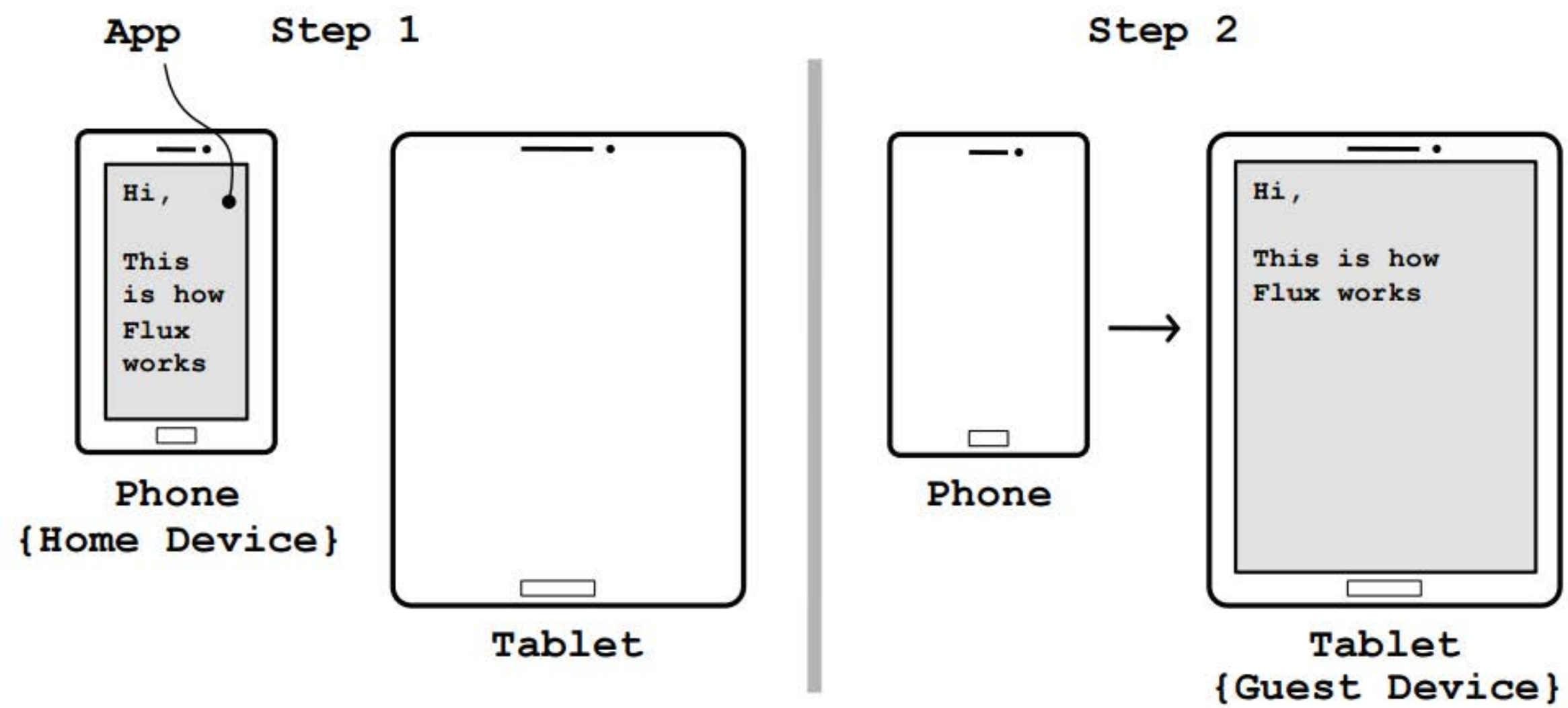
Flux

UI AppSync



App Migration

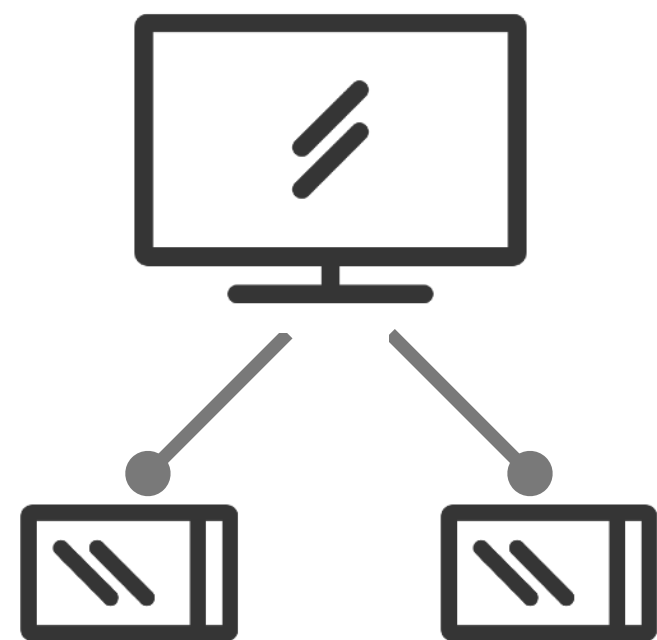
- ▶ Selective Logging of the system
- ▶ Checkpoint-Restore for Binder states



Case Study: Rio

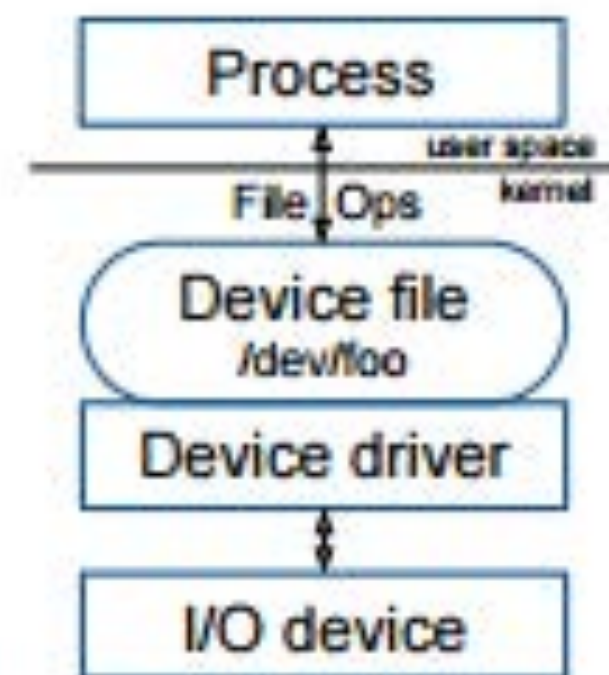
Rio

ResShare

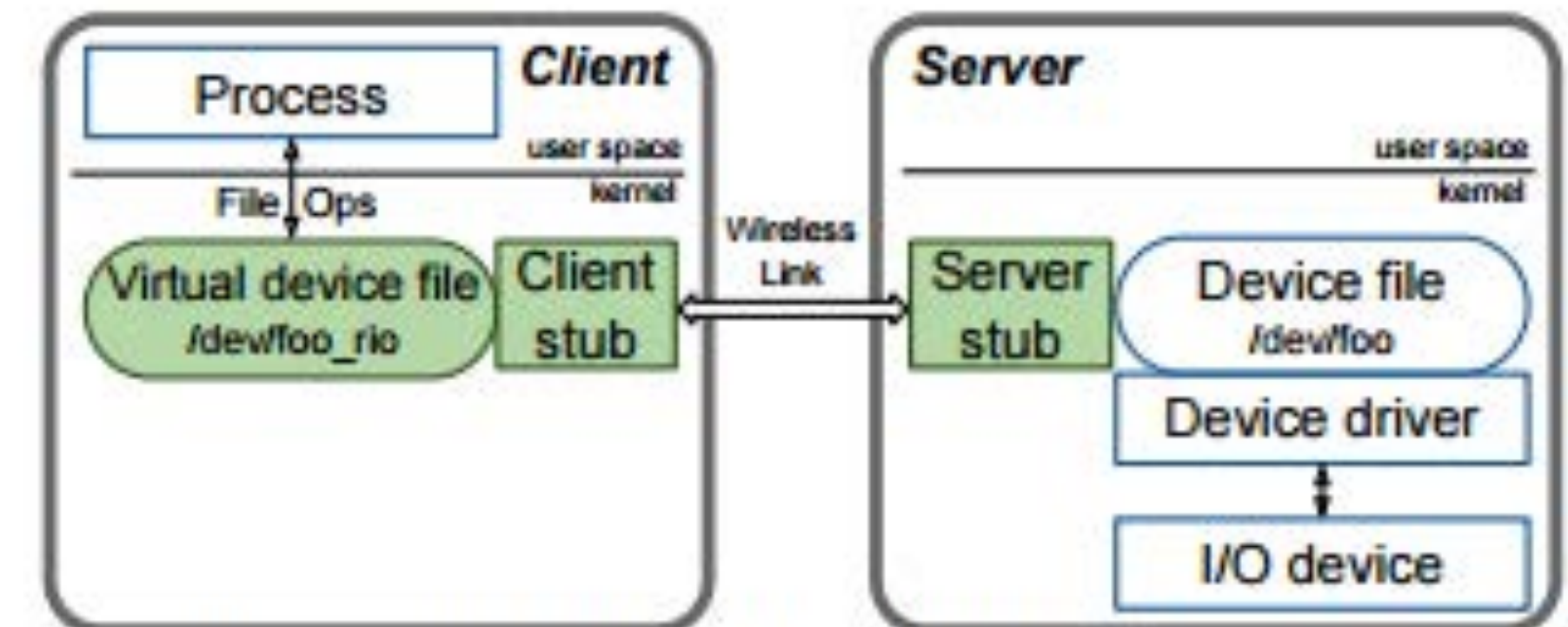


I/O sharing → Virtual device file

- ▶ Distributed Shared Memory
- ▶ Pre-copying & Batching



(a) I/O stack in Unix-like systems

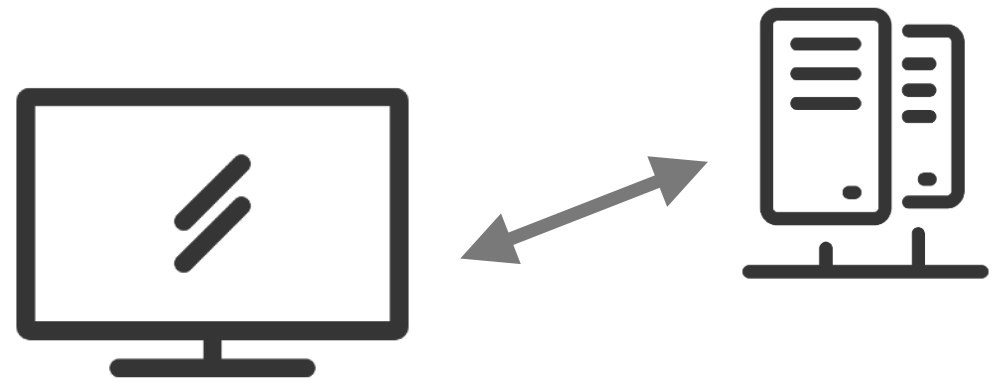


(b) Rio splits the I/O stack at the device file boundary

Flip-Flop

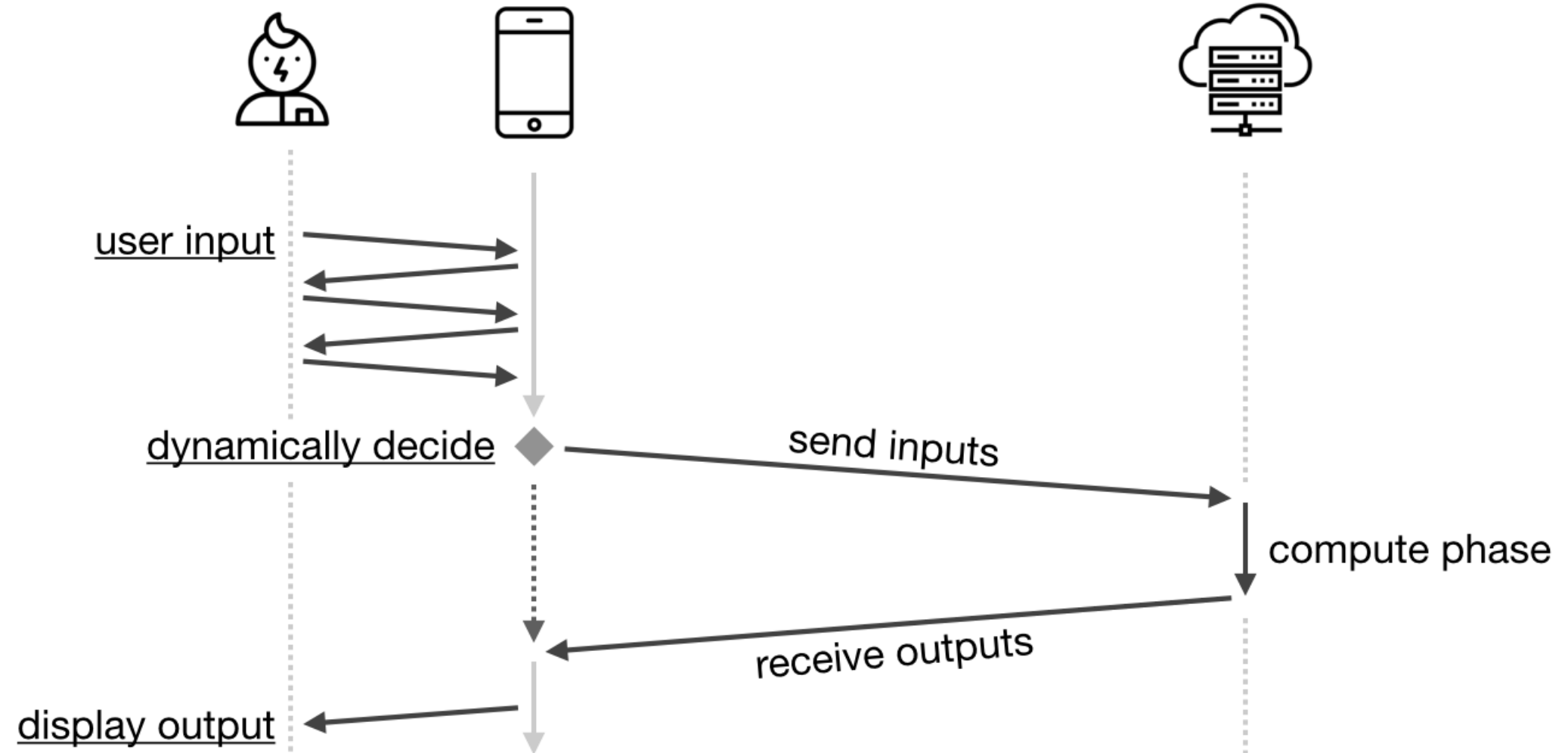
AppSync

ResShare



App Acceleration via Replication

- ▶ Replication on Dalvik VM Level
- ▶ Deterministic Replay & Leader Switching
- ▶ Server Fail Recovery



Case Study: Summary

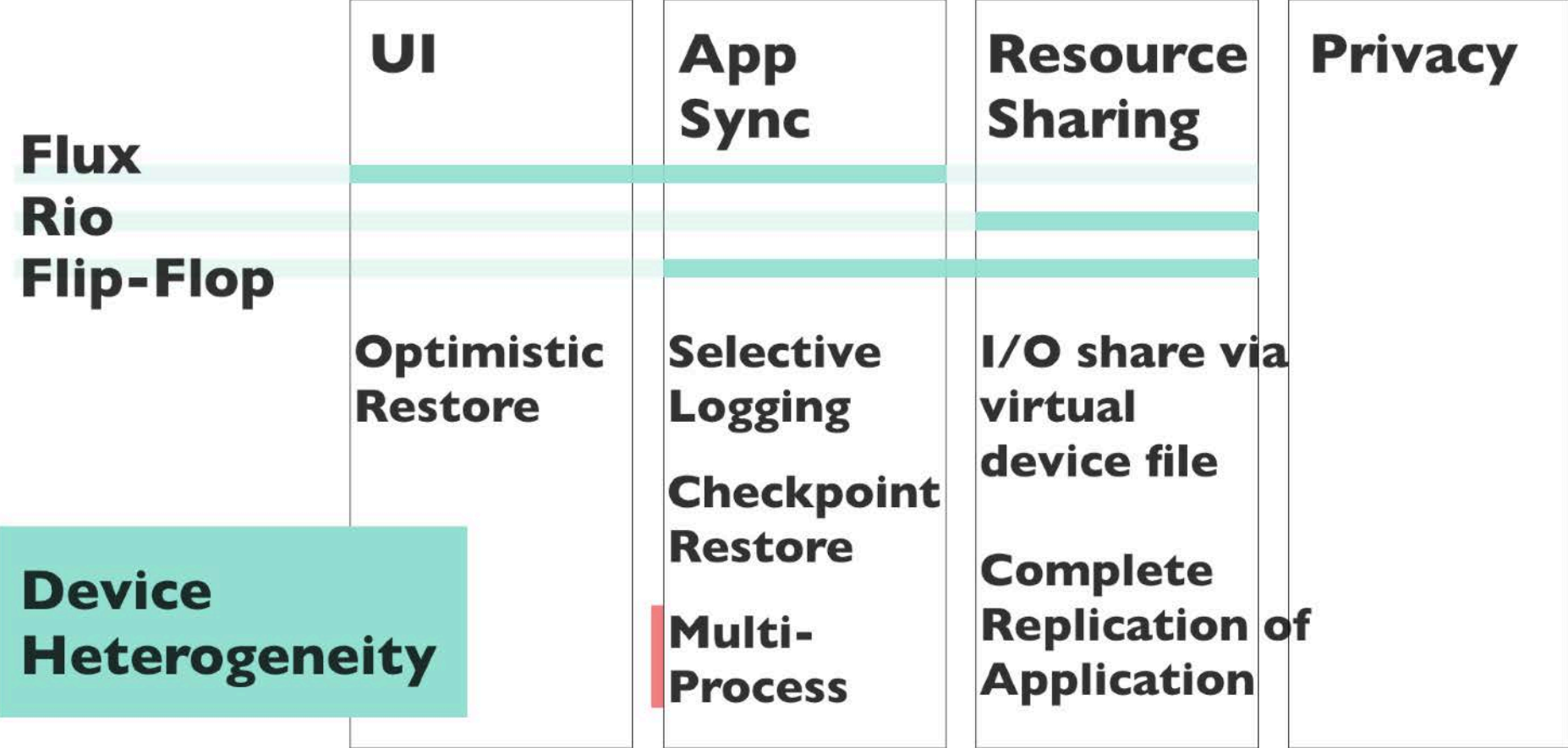
	UI	App Sync	Resource Sharing	Privacy
Flux				
Rio				
Flip-Flop				
	Optimistic Restore	Selective Logging Checkpoint Restore	I/O share via virtual device file	
Device Heterogeneity		Multi-Process	Complete Replication of Application	

Case Study: Summary

	UI	App Sync	Resource Sharing	Privacy
Flux				
Rio				
Flip-Flop				
		Migration delay	Network Acceleration	
Network			Bandwidth Problem	

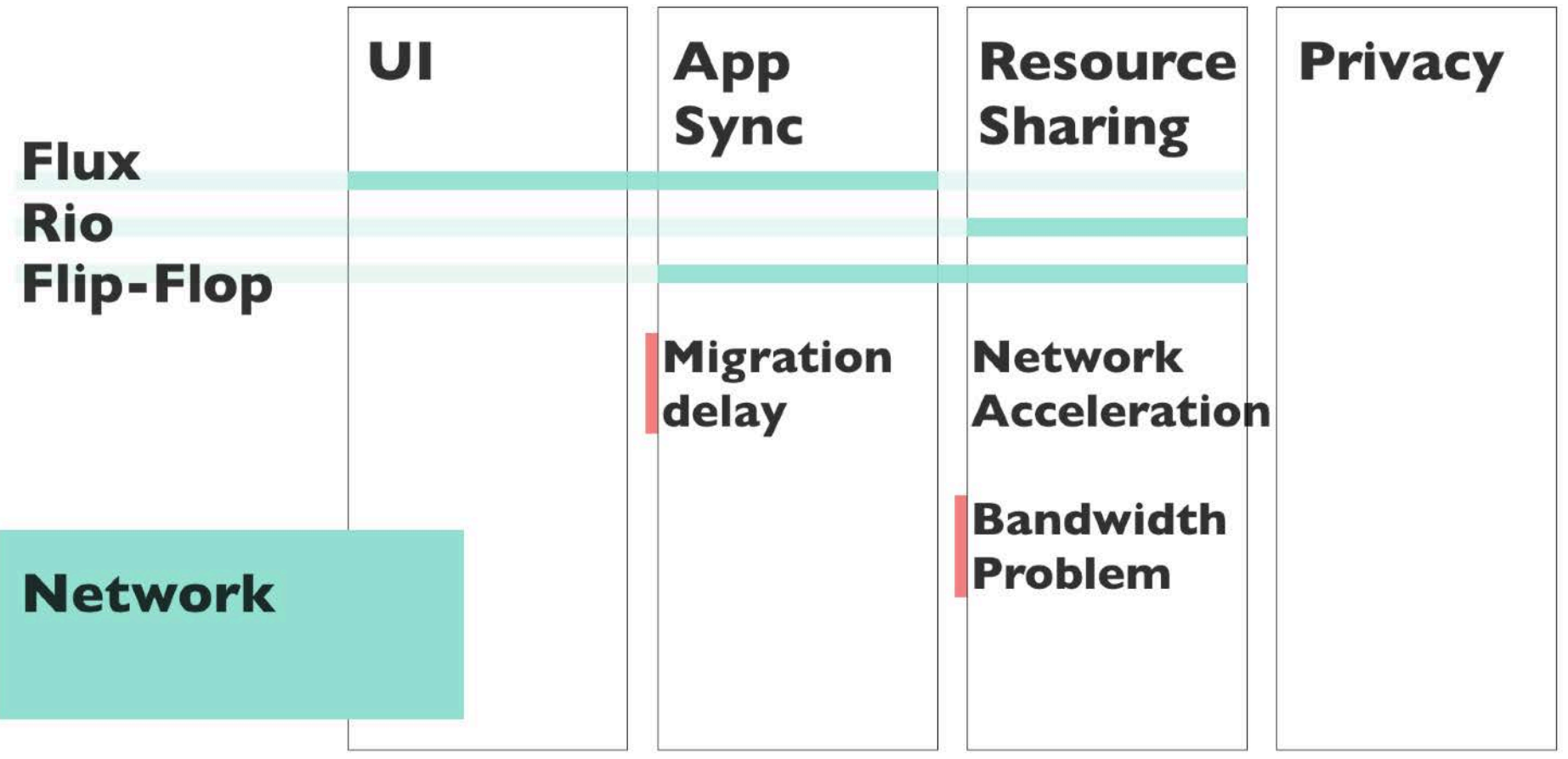
Future Directions

Device Heterogeneity



- 1. Non-determinism
- 2. Multi-Process

Network



1. Bandwidth

Compression / Congestion

2. Fault Tolerance

Complicate policy needed

Privacy

- 1. Adaptive privacy policy**
i.e. proximity, authentication strategy
- 2. Intelligent private-info detection**
i.e. one-time GPS call v.s.
frequent GPS call



any questions?