

WEEK 2

WHAT ACTUALLY HAPPENS WHEN YOU TAP A BUTTON?

A public, self-guided mini-lesson for any age or experience level.

Human action → **device signal** → **software event**
→ **visible response**

No computer required. A pencil, paper, and curiosity are enough.

TAP



input

event

intent

feedback

How to use this public deck

Move at your own pace. Stop at the depth that helps you learn.

EXPLORER

Understand what happens at a high level.

Name the major parts. Ask good questions.

BUILDER

Interact with the system.

Map a tap. Test one thing. Improve one behavior.

ENGINEER / RESEARCHER

Increase the resolution.

Add interfaces, state, timing, trust, measurements, failure domains.

Leave with one thing: an answer, an artifact, a next step, or a better question.

Quick bridge from Week 1

The screen is where we experience technology, not where it ends.

WEEK 1

**Technology is
a system, not
a screen.**

We learned to look for hidden layers beneath everyday experiences.



WEEK 2

**Follow one tap
through the
system.**

We zoom into one button press and trace how the system turns it into a response.

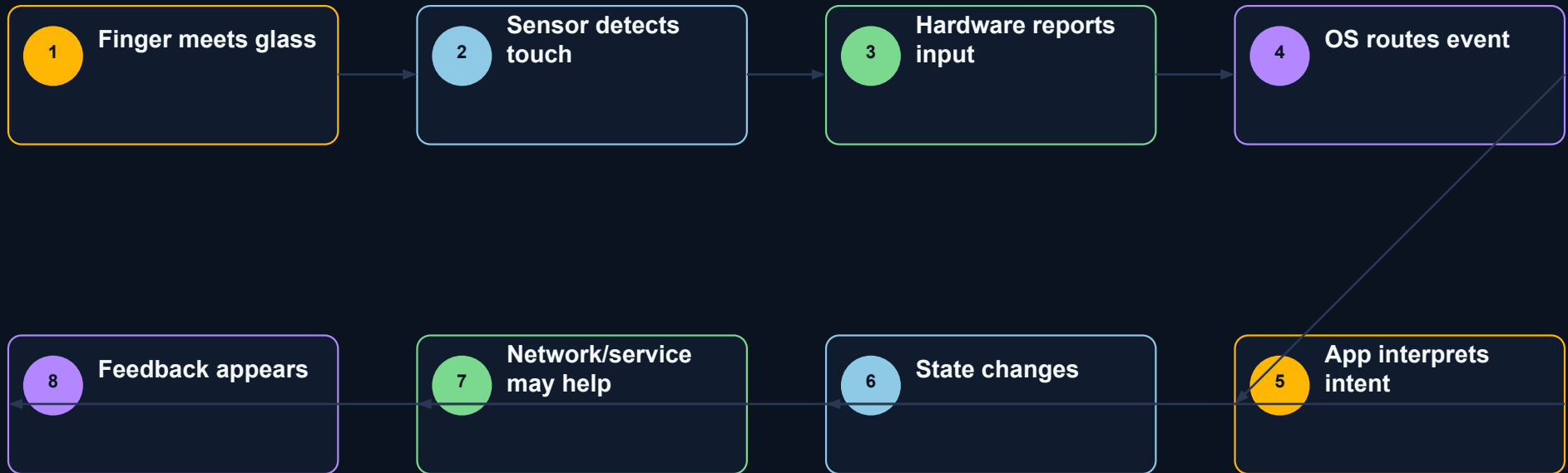
What do we mean by “tap”?

A tap is a human action that becomes a system event.



The tap journey in eight steps

You do not need to memorize these. Use them as a map.



A tap feels instant because the system does all of this fast enough that you experience one simple action.

Step 1 — Finger meets glass

The experience starts with a human intention, not with the machine.

A tap begins as a decision:

**“I want this
to happen.”**

Before there is a sensor reading, there is a person trying to accomplish something.

TRY THIS

- Pick one button you tapped today.
- What were you trying to accomplish?
- Would the same tap mean something different in another app?

Example: tapping a triangle icon might mean play a video, start a game, or expand a menu.

Step 2 — The screen detects a touch

Modern touchscreens sense a change; they do not “know” your intention yet.

The touchscreen detects where contact happened.

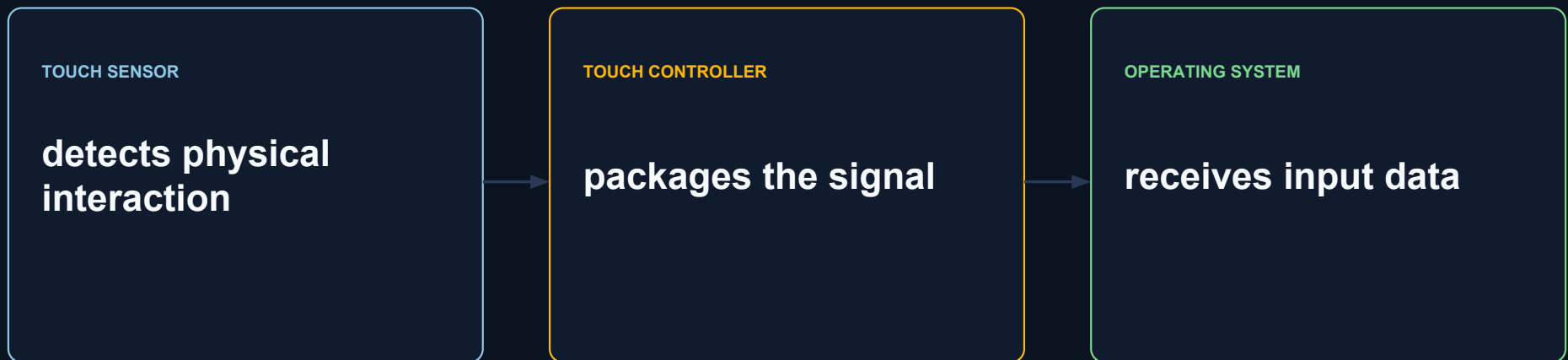
At this point the device has data like position, timing, pressure/area signals, and movement — not meaning.

Touch data, not app meaning



Step 3 — Hardware turns contact into input data

The device reports the touch to the rest of the system.



Key idea: raw hardware input must become a software event before an app can react.

Step 4 — The operating system routes the event

The OS helps decide which app and interface element should receive the tap.

Operating System

- knows which app is active
- tracks windows/screens
- handles permissions
- sends events to apps

The OS asks a practical question:

“Who should receive this input?”

This is why tapping the same physical spot can do different things depending on which app, screen, or mode is active.

Step 5 — The app interprets the tap as intent

The app maps an event to a rule: play, submit, open, like, move, buy, send.

PLAY

start or resume media

SEND

submit a message

OPEN

change screens

BUY

begin checkout

MOVE

control a game

ASK

send a prompt

The same tap can trigger different app logic depending on what the app designed that button to mean.

Step 6 — Something changes state

After the app understands the tap, the system changes.

State means:

**what the system
currently remembers
or believes is true.**

Examples: paused vs playing, empty cart vs filled cart, logged out vs logged in, unsent vs sent.

ASK YOURSELF

- What changed because of this tap?
- Where is that change remembered?
- Could it be undone?
- Does another device need to know?

Step 7 — Some taps stay local. Some taps leave the device.

This distinction explains a lot about speed, privacy, and reliability.

LOCAL TAP

The device can respond without contacting a remote service.

- Opening a downloaded file
- Toggling a setting
- Moving in an offline game

REMOTE TAP

The tap needs help from a network, service, account, or database.

- Sending a message
- Submitting homework
- Asking cloud AI
- Buying something online

If the tap leaves the device, it becomes a request

A tap can create a message that travels through a network to a service.



The response travels back. Then your app updates what you see, hear, or feel.

Step 8 — Feedback makes the system feel alive

A good interface quickly tells the human: “I heard you.”

VISUAL

button changes,
screen updates

AUDIO

clicks, alerts,
confirmation
sounds

HAPTIC

vibration,
controller rumble

STATUS

loading indicator,
success, error

If nothing changes after a tap, the user may not know whether the system ignored them, failed, or is still working.

Why taps feel fast, slow, or ignored

The problem may live in different parts of the journey.

TAP FEELS FAST

The system detects, routes, processes, and responds within the user's expectation.

TAP FEELS SLOW

The action may need network, server, database, computation, or rendering work.

TAP FEELS IGNORED

The app may not receive the event, may be busy, may lack feedback, or may have failed.

Good design makes delays understandable. Bad design makes users guess.

Troubleshoot a bad tap

Observe before you guess. Then narrow the system.

1

OBSERVE

What exactly happened after the tap?

2

LOCATE

Which part of the journey could explain it?

3

TEST

What simple check separates causes?

4

LEARN

What did the evidence teach you?

Example: If a button works in one app but not another, the screen probably is not the whole problem.

Try: tap another button, switch apps, check network, wait for feedback, restart the app, compare another device.

Activity — Map one tap

Complete this with paper, a phone note, or a whiteboard.

Choose one button you actually tapped today.

1

What did you want to happen?

2

What hardware sensed the tap?

3

Which app or screen received it?

4

What changed state?

5

Did the tap stay local or contact a service?

6

What feedback told you it worked?

If you do not know an answer, write “?” — that is a useful learning marker.

Reflect + next Thursday

Turn your map into a question, test, or project.

REFLECTION

- Which step was easiest to understand?
- Where did you write a question mark?
- What could make this tap feel slow?
- What would you improve if you designed the app?

NEXT THURSDAY

Why does technology feel fast, slow, smooth, or unstable?

We will connect human experience to timing, reliability, responsiveness, and system performance.

Bring a question. Bring a problem. Bring an idea. Or just bring your curiosity.