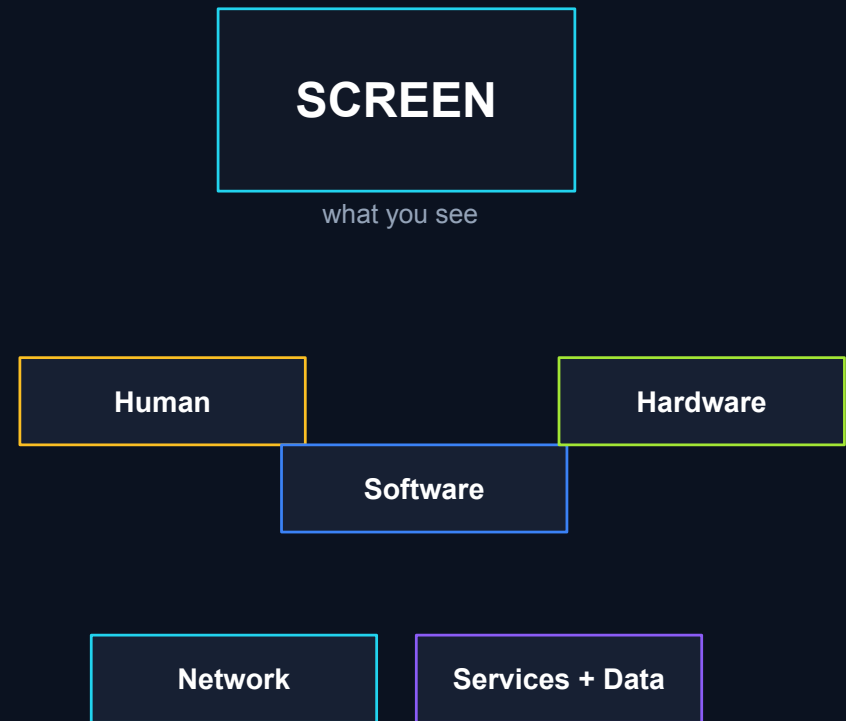


Technology Is a System, Not a Screen

Thursday Technology Open Lab · Week 1 public deck

Ask · Learn · Build



How to use this deck

Self-guided, facilitator-led, or streamed — no computer required.

Materials

Paper + pencil
Phone optional
Computer optional

Explorer
Understand the big idea
in plain language.

Builder
Draw, test, change, or
explain a system.

Engineer / Researcher
Reason about
dependencies and failure
modes.

Time

15 minutes: core lesson
10 minutes: activity
5 minutes: reflection

You do not need to “finish” every layer. Stop at the level that helps you today.

Goal: leave with an answer, an artifact, a next step, or a better question.

The screen is where the experience appears

It is not where the system ends.

A button
A video
A message
A map
An AI answer

visible
experience

Power

Input

OS

App

CPU/GPU

Memory

Wi-Fi/Cell

Server

Data

Security

hidden supporting system

Start with a familiar moment

Choose one ordinary experience. The technical system comes next.

Send a message
Who receives it? Where does it travel?

Watch a video
Why does it buffer or play smoothly?

Use GPS
How does the map know where you are?

Ask AI a question
Where does the answer come from?

Play an online game
Why does lag change the experience?

Buy something online
What has to be trusted?

Activity preview: pick one moment and draw what must exist for it to work.

What you see / what you do not see

A simple experience can depend on many invisible parts.

WHAT YOU SEE

Tap
Screen changes
Sound plays
Message sends
Video moves

WHAT SUPPORTS IT

input

application

memory

network

database

operating system

processor

storage

server

security

Example: press Play on a video

A one-second experience can cross many boundaries.



Human experience: “I tapped Play and the video started.”

System view: input → software → device → network → service → output

The five-layer model

Use this as your first map for almost any technology experience.

1 · HUMAN

What is somebody trying to do?

2 · SOFTWARE

What instructions, interface, and logic create the experience?

3 · HARDWARE

What physical device performs the work?

4 · CONNECTION

How does information move between systems?

5 · SERVICES + DATA

What lives somewhere else that the experience depends on?

Use the model like a flashlight: it helps you see what is usually hidden.

Layer 1: start with the human

Purpose comes before parts.

Ask first:

What is the person trying to do?

Good technology preserves the experience people care about.

If the human outcome fails, the system failed somewhere — even if a status light is green.

Student

“I need to submit my work.”

Parent

“Can I understand this app?”

Gamer

“Can I play without lag?”

Builder

“Can I make this better?”

Layers 2–3: software + hardware

Software gives instructions. Hardware performs physical work.

SOFTWARE
apps
operating system
code
interfaces
permissions

instructions
become actions

HARDWARE
screen
processor
memory
storage
radio
battery

Beginner check: Can you name one app and one physical part involved in your chosen experience?

Layers 4–5: connection + services/data

Some parts of your experience may live far away.

YOUR DEVICE
app · OS · hardware

CONNECTION
Wi-Fi · cellular ·
internet

REMOTE SERVICES
servers · data ·
accounts

Examples: cloud storage, streaming video, multiplayer game servers, AI systems, online banking, school portals.

Question: Which parts of your chosen experience still work without internet?

Connected does not always mean usable

A system can be “online” while the human experience has already failed.

Wi-Fi connected $\neq$ **video working**

Latency

Reliability

Bandwidth

Power

Compute

Server health

Permissions

Software state

Technology succeeds when enough hidden conditions stay within acceptable bounds.

Troubleshoot without guessing

Narrow the system before you name the cause.

1

OBSERVE
What exactly
happened?

2

LOCATE
Where might the
problem live?

3

TEST
What simple check
separates two
explanations?

4

LEARN
What did the evidence
change?

Example: One video fails? All apps fail? One device fails? Whole building fails?
Each answer points somewhere different.

Activity: draw one technology experience

No computer required. Use paper, a notes app, or a whiteboard.

Step 1

Write one experience in the center of the page.

Examples: send a message · watch a video · use GPS · ask AI · play a game · buy something online



Step 2: connect your center experience to all five layers.

Choose your challenge level

Everyone completes the same activity at a useful resolution.

Explorer

Put at least one item in each layer.

**Human · Software · Hardware ·
Connection · Services/Data**

Builder

**Add one thing you could test,
change, fix, or build.**

What could you improve?

Engineer / Researcher

**Add interfaces, state, trust
boundaries, bottlenecks,
measurements, and failure
domains.**

**What evidence would prove your
claim?**

Rule: do not stop because you are unsure. Mark unknowns with “?” — unknowns are where learning begins.

Reflect: what did your map reveal?

The goal is not a perfect drawing. The goal is better questions.

What part was easiest to identify?

What part was invisible until today?

What would fail first?

**What would you like to learn, test, or
build next?**

**Leave with one: answer · artifact · next step · better
question**

Optional deep cut: turn boxes into a system

For builders, engineers, researchers, and curious overachievers.

Replace each arrow with a question:

What crosses this boundary?

Where does state live?

What is trusted here?

What happens on retry?

What would the user notice first?

What measurement would prove it?

component

interface

A diagram becomes architecture when it explains dependencies, evidence, and failure.

Week 1 takeaway

The screen is where you experience technology. It is not where technology ends.

Whenever technology looks simple, ask:

What has to be true for this experience to work?

**Next week:
What actually happens when you
tap a button?**

**Continue learning:
WAIKE + The Technology Landscape
github.com/gunnchOS3k/waike-research-ops
github.com/gunnchOS3k/gunnchos-technology-landscape**

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