Edge, fog and mist

You tap your card on the bus validator and it beeps at once, even in a tunnel with no signal. Your phone unlocks with your face in a fraction of a second, also in airplane mode. Neither of them asked a data centre in Madrid or Frankfurt for permission. Why not, if the cloud is so powerful?

Because the cloud is far away, and distance costs time. A wind turbine that must stop when a gust becomes dangerous, a pump that must stop before a pipe bursts, a hotel door that must open when the internet is down: none of them can wait. Today you measure that distance with your own browser, and you learn where the decisions that cannot wait are taken.

What you will learn

- Measure **latency** to real data centres and explain why it limits what the cloud alone can do
- Describe **edge computing** and how it works **with** the cloud, not against it
- Define **fog** and **mist** and place the four layers in order, from the sensor to the cloud
- Decide, in a real plant, which task goes in which layer and why
- Avoid the trap: “edge replaces the cloud”

How far away is the cloud?

> **Latency** is the delay between sending a request and receiving its answer. It depends mainly on the distance and on the number of networks the data cross. It is measured in milliseconds (ms).

You can measure it right now. GCPing opens in the browser, with no account, and measures how long your computer takes to get an answer from each Google Cloud region. Look for Madrid (`europe-southwest1`), then for a region in the United States and one in Asia. The figures change every day and from one network to another, so trust the ones you see, not a figure in a book. The pattern is always the same: the further away, the longer it takes.

For a film, a few dozen milliseconds do not matter. For a machine that has to react, they can. And if the connection drops, the cloud is not slow: it is simply unreachable.

Edge computing: decide close to the data

> **Edge computing** means processing data **close to where they are produced** (in the device itself, in the same room or in the same building) instead of sending everything to a distant data centre.

In the city-builder game, the cloud is the utility’s big power station at the other end of the map. You would never ask it to switch each traffic light. So you install **local substations** near the neighbourhoods, and they handle what must happen now.

	Edge	Cloud
Where	Next to the devices: a gateway, a small local server, the device itself	The provider's remote data centres
Latency	Very low	Higher: depends on distance and network
Power and capacity	Limited	Very large and expandable
Without connection	Can keep working	Cannot be reached
Data	Filters and summarises them; sends only what is needed	Receives the summary, stores it and analyses it in depth
Typical tasks	React at once, filter, decide locally	History, global analysis, training models, coordinating many sites

Why not send everything to the cloud? Four reasons: **latency**, **bandwidth** (sending everything costs money and network), **working offline**, and **privacy** (some data are better never leaving the building).

You already carry edge computing in your pocket. Apple says the Face ID data never leave the iPhone: the face is checked inside the phone. Your photos, on the other hand, sync with the cloud. Websites use edge too: a **CDN** (*content delivery network*) keeps copies of a site on servers spread around the world, so each visitor gets it from a nearby point. Your GitHub Pages site is served that way.

> **Careful: "edge replaces the cloud"**
> It does not. They share the work. The edge **reacts** and **filters**; the cloud **remembers**, **analyses** and **coordinates**, and sends back down what it has learnt: new rules, settings, updates. Fast decisions at the edge; smart decisions in the cloud.

Fog and mist: the layers in between

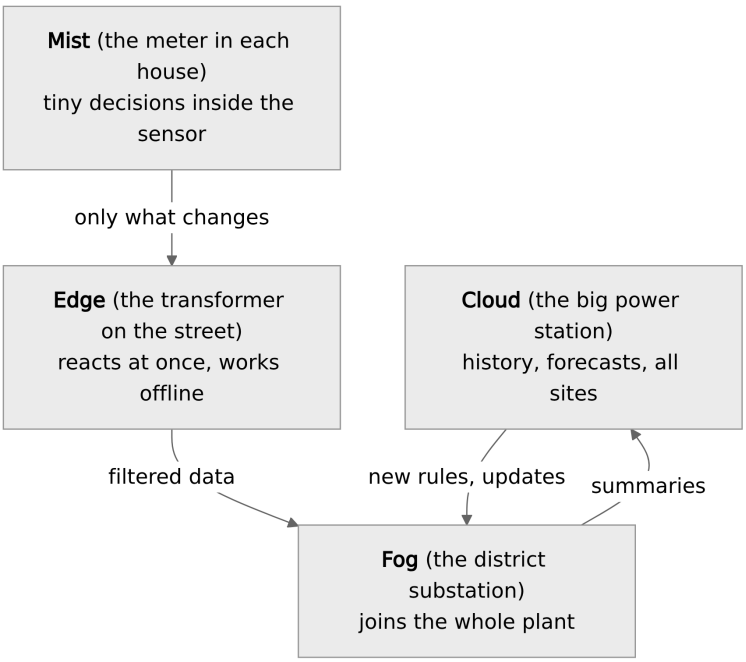
Between the sensor and the cloud there can be more than one step.

> **Fog computing** is an intermediate layer on the local network, between the edge devices and the cloud, that joins and processes the data of many points before deciding what goes up. **Mist computing** is the lightest layer, inside the sensors and actuators themselves, on very low-power microcontrollers.

A way to remember it: the cloud is high in the sky; **fog** is a cloud touching the ground; **mist** is even thinner and right at ground level. The lower you go, the closer to the sensor, the less power and the faster the answer.

The same four layers in three real kinds of plant (these are teaching examples, not the description of one particular installation):

Layer	Where it lives	Wind farm	Water utility	Hotel
Mist	In the sensors and actuators	A vibration sensor on a turbine only reports when it crosses a threshold	A flow meter sends a reading every few minutes, or at once if it jumps	A room temperature sensor only reports changes
Edge	Next to the devices	Each turbine's controller turns the blades and	The pumping station's controller stops the pump if pressure spikes	Each floor's controller opens doors and regulates air conditioning



Layer	Where it lives	Wind farm	Water utility	Hotel
		stops it in dangerous wind		
Fog	On the local network of the site	The farm's substation server joins all turbines	The control centre joins all stations of an area	The hotel's server joins all floors and manages the building's energy
Cloud	Remote data centres	History of all farms, production forecasts, predictive maintenance	Consumption of the whole city, leak detection by area, reports	The chain compares all its hotels and predicts occupancy

Different sources, different borders

The order **mist → edge → fog → cloud** is the convention of this module. Not all sources agree. NIST's *Fog Computing Conceptual Model* (SP 500-325, 2018) says there is no consensus: it places fog between the end devices and the cloud and mist as a lighter form of fog close to the devices, but it calls **edge** the layer of the end devices themselves. If you read it that way somewhere, you are not wrong. What matters is that you can order the layers and say **what each one does**.

In class: where do we process it?

Individually or in pairs. Use the plant of your project company (the OT side of D1). If it has none, choose one of the three in the table: a wind or solar farm, a town's water utility (for example EMAYA in Palma) or a hotel. **No generative AI**: this activity is assessed (class work, 30%).

Step	Do this	You should get
1	Measure latency with GCPing: note the fastest and the slowest region, with their milliseconds	Two figures and a date

Step	Do this	You should get
2	**Guided research:** find **one real source** that says which equipment, sensors or control systems this plant (or one like it) has: company website, press release, annual report, public tender, manufacturer sheet	One citation: title, organisation, link, date consulted, and the sentence that says it
3	Write **six tasks** done with the plant's data (stop a machine, filter readings, join the whole plant's data, monthly report, predict breakdowns…)	Six tasks
4	Place each task in **mist, edge, fog or cloud** and give the **reason**: latency, working offline, data volume, privacy or computing power	A table of six rows
5	Name **one piece of data you would never send to the cloud**, and why	One sentence
6	Draw the four layers and the arrows of the data flow with diagrams.net or Mermaid Live (no account) and export it as an image	A diagram

Hand in on Classroom the citation, the table and the diagram.

For your project

This lesson feeds Deliverable 3 · Cloud architecture:

Today's work	Becomes	Criterion
The six tasks in their layers, with reasons and the real source	Section C: edge, fog and mist in the plant	c, d
The diagram of the four layers	The base of the group's architecture diagram	Group part

Key takeaways

- **Latency** is the delay between a request and its answer; it grows with distance, and without a connection the cloud cannot be reached at all.
- **Edge computing** processes data close to where they are produced, to react fast, work offline, save bandwidth and protect privacy.
- Edge and cloud **share the work**: the edge reacts and filters; the cloud remembers, analyses and coordinates.
- **Fog** is the intermediate layer that joins a whole site; **mist** lives inside the sensors themselves.
- In this module, from the sensor up: **mist → edge → fog → cloud**. Sources draw the borders differently; what counts is what each layer does.

Quick check

1 What is latency?

- **A** The amount of data a connection can carry per second
- **B** The delay between sending a request and receiving its answer
- **C** The time a server stays switched on
- **D** The distance in kilometres to the data centre

2 What is edge computing?

- **A** Moving all processing to the biggest data centre available
- **B** Processing data close to where they are produced
- **C** A type of SaaS for small companies
- **D** Storing backups on the border between two countries

3 A wind turbine must stop at once if a gust becomes dangerous, even if the internet is down. Where should that decision be taken?

- **A** In the cloud, which has more computing power
- **B** At the edge, in the turbine's own controller
- **C** In the provider's data centre closest to the farm
- **D** It does not matter, as long as the data are saved

4 Following this module's convention, which is the order from closest to the sensor to furthest away?

- **A** Edge → mist → cloud → fog
- **B** Fog → mist → edge → cloud
- **C** Mist → edge → fog → cloud
- **D** Cloud → fog → edge → mist

5 Which statement is true?

- **A** Edge computing replaces the cloud completely
- **B** Filtering data at the edge reduces the volume sent to the cloud
- **C** Mist computing runs in the provider's data centre
- **D** Fog computing is the same as a CDN

6 Which task fits best in the cloud?

- **A** Opening a hotel room door when the guest taps the card
- **B** Stopping a pump when pressure spikes
- **C** Comparing the energy use of all the hotels of a chain over five years
- **D** Sending a temperature reading only when it changes

7 A supermarket chain's cash registers must keep selling when the connection fails, and head office wants to know the sales of every shop each night. Which design fits?

- **A** Everything in the cloud, because it is cheaper
- **B** Everything in each shop, with no cloud at all
- **C** Each shop sells and records locally (edge) and sends the day's summary to the cloud for head office
- **D** Every sale waits for the cloud to approve it

You now know where each decision is taken. One question is left: whose power station is it, and which law applies to what it stores? Next: Advantages and sovereignty.

Solutions

1. B · **Latency** is a delay, measured in milliseconds. It depends mostly on distance and the networks crossed; how much data fits per second is **bandwidth**.

2. B · **Edge computing** processes data near their source (device, room or building) so the answer arrives fast and works without a connection.

3. B · The decision cannot wait for the network, and must work **offline**: it belongs at the **edge**.

4. C · **Mist** inside the sensor, **edge** next to the devices, **fog** on the site's local network, **cloud** far away.

5. B · The edge **filters** and **summarises**, so far less data travel up. The cloud is still needed to store, analyse and coordinate.

6. C · Long history and many sites need **storage and computing power**: that is the cloud's job. The other three are edge or mist tasks.

7. C · The same split as the wind farm: **react locally** so the shop never stops, and **send summaries** up so the cloud can compare all shops.