Engineer

Dashboard

one screen built to answer a recurring question

What it is

A dashboard is a page of panels, each one a query turned into a picture, laid out so that someone can read the state of a thing without asking anyone. The same idea goes by other names: Prometheus's own documentation calls them consoles, and a team might say "the board" or "the screen on the wall". Grafana describes its dashboards as one or more panels arranged in rows or tabs that "provide an at-a-glance view of related information", and each panel as the container for a visualisation built from a query against a data source. That source can be a time-series store, a SQL database, an API or a plain CSV file.

The panels come in a handful of shapes. A line over time shows a trend: memory in use across the last day, requests per minute across the last week. A single large number answers "how much right now". A gauge shows a value against a known range, such as disk used out of disk available. A table lists things that need attention, one row each. A status light says up or down. Most good dashboards use two or three of these shapes, picked for the question, and very few panels of each.

On a web page those panels are ordinary interface parts, and the W3C's ARIA Authoring Practices Guide describes how each should behave for people using a keyboard or a screen reader. Tabs switch between views and show one panel of content at a time. A table is a static grid of rows and cells; a grid is the interactive version, moved through with the arrow keys. A meter is the pattern for a value inside a defined range, and an alert is a short, important message that draws attention without interrupting what the reader is doing. A gauge written to the meter pattern carries its range and its reading as text:

```
<div
    role="meter"
    aria-label="Disk used"
    aria-valuemin="0"
    aria-valuemax="100"
    aria-valuenow="72"
    aria-valuetext="72% (31 GB free)"
></div>
```

A dashboard waits to be looked at, and that sets it apart from its neighbours. An alert comes to a person when something crosses a line. A report is a snapshot, produced when asked for and then fixed. A dashboard is live, and it is worth exactly as much as the attention someone gives it.

Use the concept in a decision

Put “What is a dashboard?” to work in a real decision. We can help relate the concept to the system you are building or buying. Dashboards and reports showing the handful of figures a decision actually turns on, refreshed without being asked.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Brand communication informed by audience response — brand communication informed by audience response

Brand communication informed by audience response Get in touch →

Why it matters

A dashboard gives a team one shared reading of the state of things. Without it, the same question gets answered by whoever has a terminal open, in their own way, and two people in an incident can spend minutes discovering that they were looking at different numbers. With it, the conversation starts from the same picture, and people who would never write a query can see how the system is doing.

It also shortens the walk from “something is wrong” to “this part is wrong”. The Prometheus guidance on consoles recommends building one dashboard per service, each showing the latency and errors of the services it talks to, so that someone can start at the top of the tree and follow the bad numbers down to the service causing them. A dashboard laid out along the structure of the system turns that walk into a few clicks.

It has a cost too. Every panel runs its query on every refresh, on every screen that has it open. Every panel is also a claim that someone keeps true when the service behind it is renamed, split or retired. A dashboard with forty panels has forty of those claims, and most of them go unread.

What it looks like in practice

In an earlier role, the founder built an organisation's first observability dashboard, which started by collecting CPU, RAM, disk and temperature from remote Linux servers over SSH, Docker containers included. A second version moved to Grafana and Prometheus, and the panels on it were chosen with the DevOps and engineering teams: CPU utilisation, memory usage, service uptime and API latency. It ran on the large TV in the office, where anyone walking past could see the state of production, and alerting on threshold breaches went alongside it. Teams detected and resolved incidents 40% faster.

A later monitoring dashboard on Grafana and Prometheus names the part that mattered most: metrics that meant something, rather than vanity numbers that look busy and tell nothing. Alert thresholds were then set on those metrics and surfaced where the team would see them while there was still time to act. The team moved from firefighting to catching problems while they were small.

On a host with 464 MB of memory, the same kind of question gets a different answer. Seven read-only roles render the live host into Markdown reports on demand, covering access, traffic, metrics, security and more, because a metrics stack would not fit and would not repay its cost if it did. Their governing rule suits any dashboard just as well: print only numbers the run measured, since a report that fills a gap with a plausible figure is worse than one that leaves it blank.

Where it goes wrong

A dashboard can look healthy and still mislead, and most of the ways it does come from what was put on it.

Every available number goes on the screen. It is tempting to show everything a system can measure. The Prometheus guidance warns that this produces consoles so dense that even an expert struggles to read them, and suggests at most five graphs on a console and at most five lines on each graph. Start from the likely failures and show what tells them apart.

Vanity numbers take the space. A total of requests served since launch only ever rises, and a count of registered users says little about today. They look busy and change no decision. Give the panel to a number that would make someone act if it moved.

One dashboard tries to serve everyone. What an on-call engineer needs at night, which is what is broken, differs from what a developer needs while building a feature, such as how many people hit a particular edge case. The Prometheus guidance says plainly that one set of consoles rarely serves both, and two sets work better.

Colour carries the whole message. A row of red and green dots tells nothing to someone who cannot tell red from green, nor on a monochrome screen. WCAG 2.2 success criterion 1.4.1, at Level A, says colour must not be the only visual way of conveying information; add a word, a shape or a position. Under 1.4.11, at Level AA, the parts of a chart needed to understand it need a contrast ratio of 3:1 or more against the colours beside them, which thin pastel series on a white background often miss.

A gauge implies a ceiling that is absent. The APG meter pattern is for a value within a defined range, and its own note warns against using one for a value with no meaningful maximum, or for progress. Requests per second has no natural top, so a dial for it invents one and makes an ordinary busy hour look like an emergency. Use a line over time instead.

The screen waits for a viewer who has gone home. A dashboard on an office TV at three in the morning is watched by nobody. Problems that need a person belong in alerts, and the Prometheus alerting guidance recommends alerting on the symptoms users feel and linking each alert to the console that helps find the component at fault. The alert summons someone, and the dashboard is where they land.

Further reading

- Grafana: Dashboards — panels, rows, data sources, queries and transformations, as one widely used tool defines them.
- WAI-ARIA Authoring Practices Guide: Patterns — how tabs, tables, grids, meters and alerts should behave for keyboard and screen-reader users.
- Prometheus: Consoles and dashboards — the five-graph, five-line guideline and the case for one audience per set.

- Prometheus: Alerting — alerting on symptoms, and linking each alert to the console that explains it.
- Understanding WCAG 2.2: Use of Color — why colour alone cannot carry a status, with examples.
- Understanding WCAG 2.2: Non-text Contrast — the 3:1 contrast that chart lines and interface parts need.

Glossary

Choose for your actual constraints

Bring a representative dataset, the questions it must answer and the current bottleneck. We can agree how to measure quality, throughput and recovery.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is a dashboard?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/dashboard/