Web Analytics Streaming with OpenTelemetry

Steffen Renz
Senior Software Engineer @Porsche Digital

will talk about

💥 the challenge - (web) analytics

🦄 the opportunity - OpenTelemetry

🚀 the solution - data streaming

web analytics - the good and the bad

💩 tracking

- identify users across domain
- opt: send customer behavior data to 3rd party providers
- opt: global profiles to provide personalized ads
- potential privacy nightmare

👼 app performance

- measure avg customer success
- improve UX
- improve portfolio
- filter threats and misuse
- a/b testing

Problem: Both disciplines are tightly coupled

web analytics - expect to lose more insights

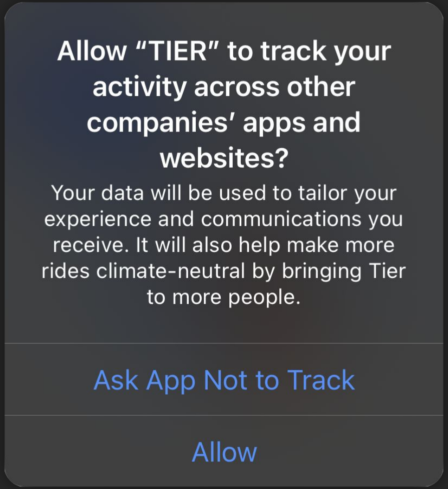
👉 EU EPD (ePrivacy directive)

👉 Apple (Intelligent Tracking Prevention)

👉 Mozilla (Enhanced Tracking Protection)

🙈 Google ("Don't be evil")

"Cookiegeddon"



Allow "TIER" to track your activity across other companies' apps and websites?

Your data will be used to tailor your experience and communications you receive. It will also help make more rides climate-neutral by bringing Tier to more people.

web analytics - architectures

👉 client-side

- applications or code executed on the client to measure usage and behavior (e.g. Google Tag Manager)
- optional: advanced trackers (cookie, fingerprinting) on client device
- allows advanced analytics like heat maps

👉 server-side

- usage and behavior is measured via client-server interactions or other backend events
- optional: group by connection data or auth

software must fit organisations

> *“aim or goal, as individuals,*
> *is to improve our capacity for independent action”*

John Boyd, Destruction and Creation, 1976

💡 idea: create analytics events out of trace spans

👉 observation: tracing and web analytics data have many similarities

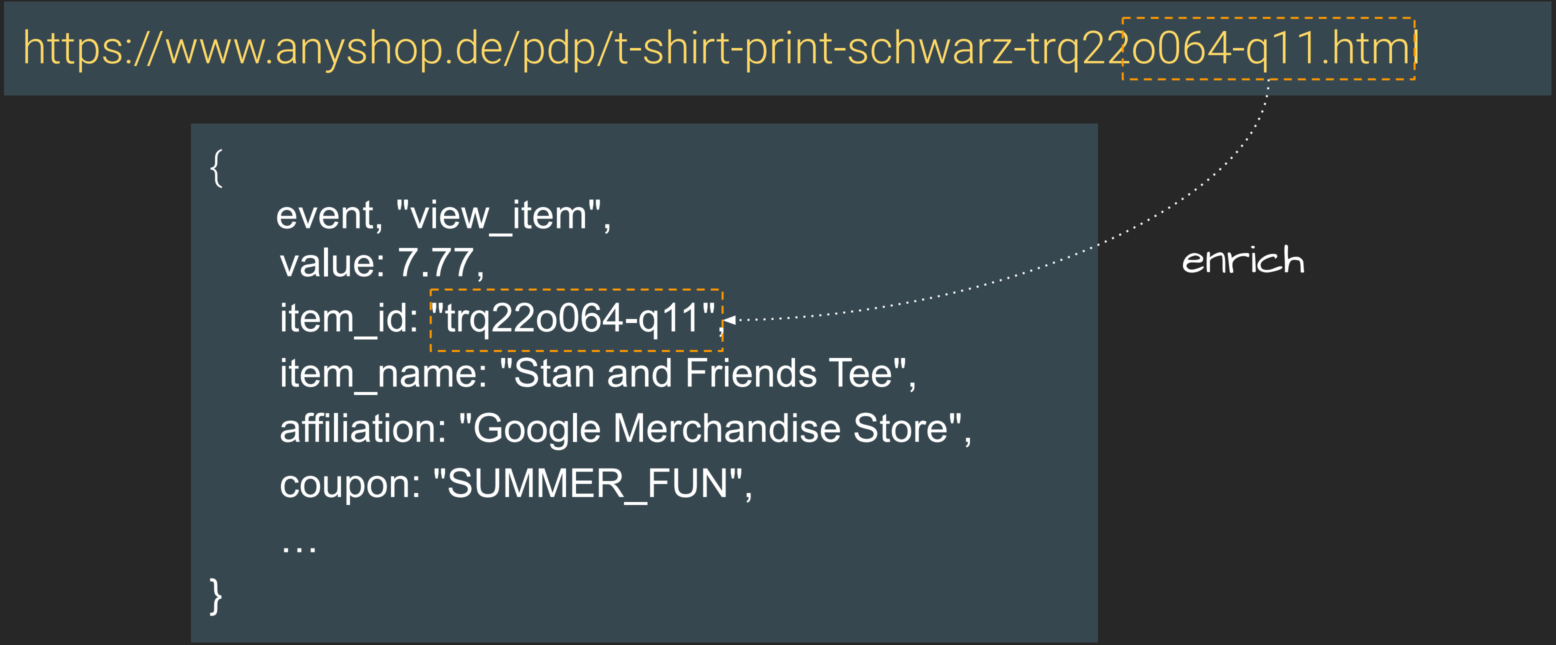
👉 expectation: APM via tracing is present in critical customer application

👉 expectation: server-side rendering will lead to more measurable calls

👉 fact: observability stacks scale well (tech & teams)

👉 fact: OpenTelemetry increases interoperability massively

from api call to json



https://www.anyshop.de/pdp/t-shirt-print-schwarz-trq22o064-q11.html

```
{
      event, "view_item",
       value: 7.77,
       item_id: "trq22o064-q11",
       item_name: "Stan and Friends Tee",
       affiliation: "Google Merchandise Store",
       coupon: "SUMMER_FUN",
       …
}
```

the opportunity

OpenTelemetry

OpenTelemetry **Protocol**

💪 traces, logs, metrics

💜 open-source

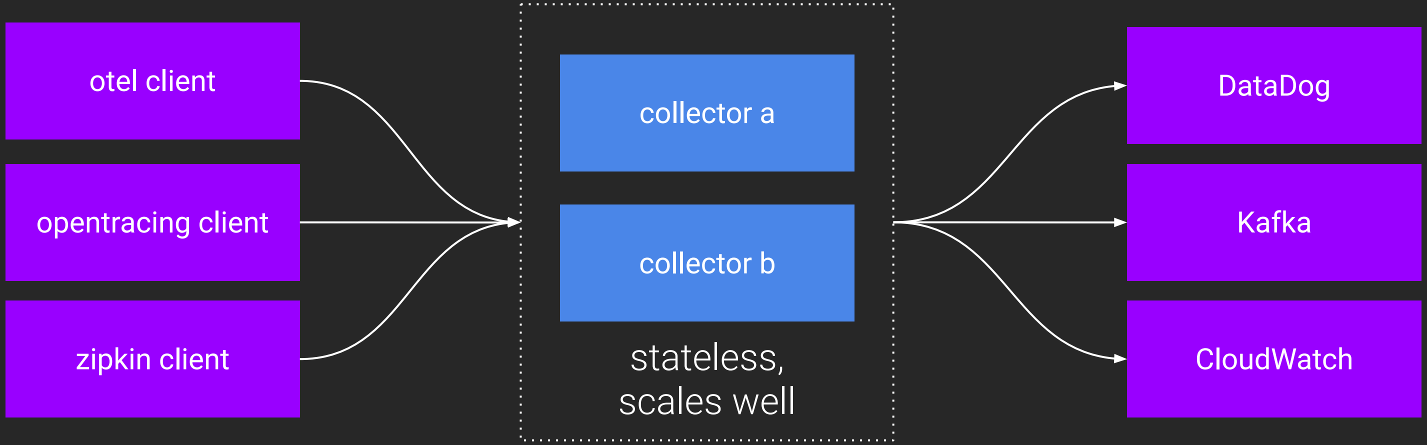
💜 bottom-up (standard through adoption)

💪 supported by community & industry

👉 replaces older protocols like zipkin, opentracing, jaeger

👉 deep dive: “Practical introduction to OpenTelemetry tracing”, Nicolas Fränkel 🇫🇷

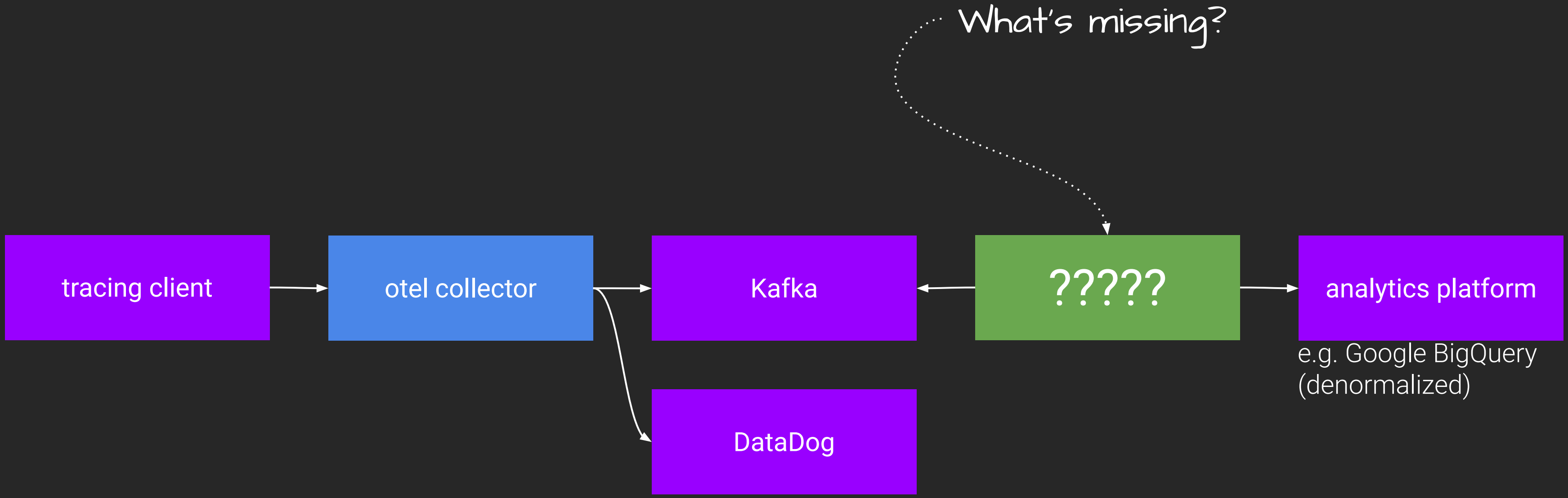
OpenTelemetry **Collector**



Receiver → Processor → Exporter

(Connector = Receiver + Exporter)

our first architecture? 🤔



e.g. Google BigQuery
(denormalized)

the solution

Data Streaming

with Kafka Streams

assumptions for larger setups

👉 disk is cheap. plenty of resources

👉 cpu is no issue. scales well

👉 network is expensive

👉 horizontal scaling, abstraction needs network

👷 We need to optimize network aka data flows -> Preferred option: Streaming

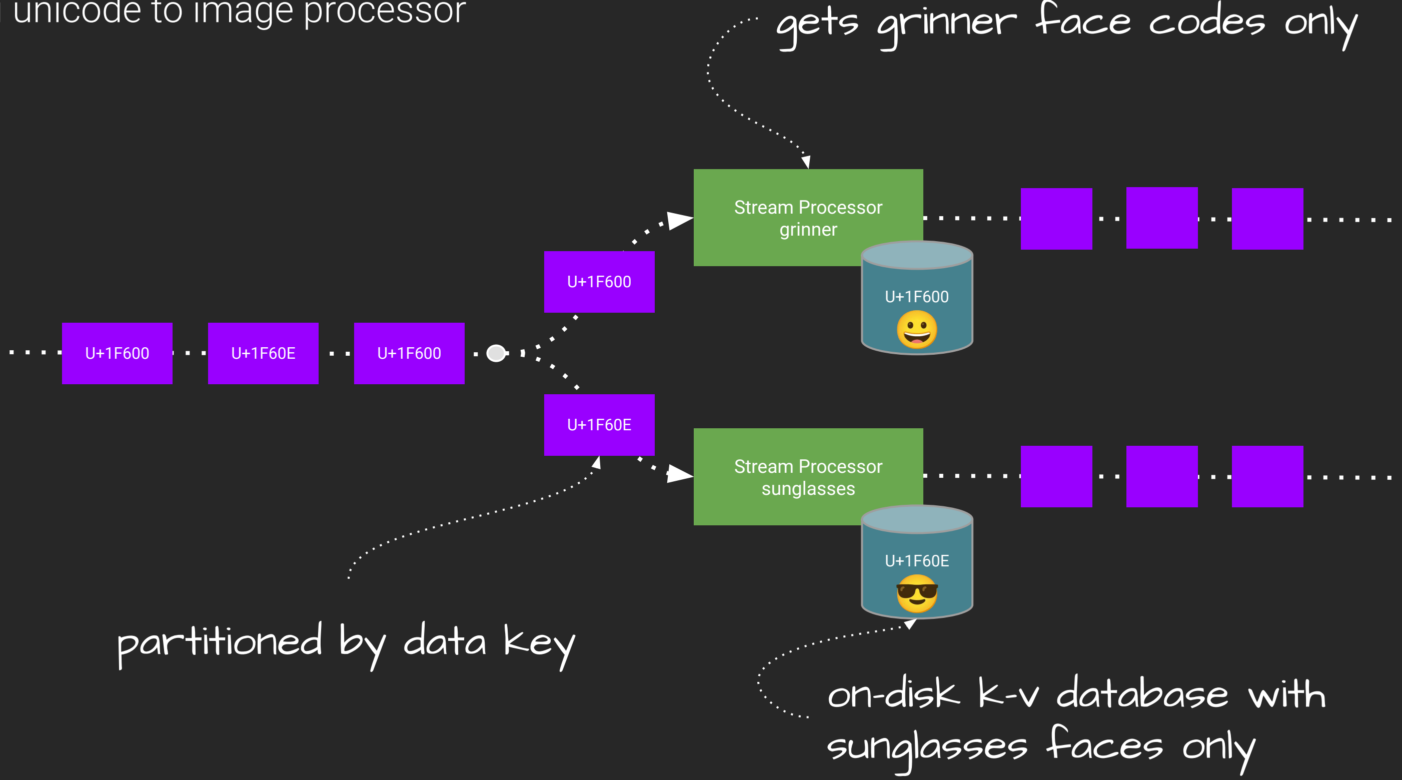
car manufacturing - an optimized stream

Porsche AG, Zuffenhausen

- 👉 everything serves main line
- 👉 parts in place when needed
- 👉 short ranges
- 👉 throughput essential

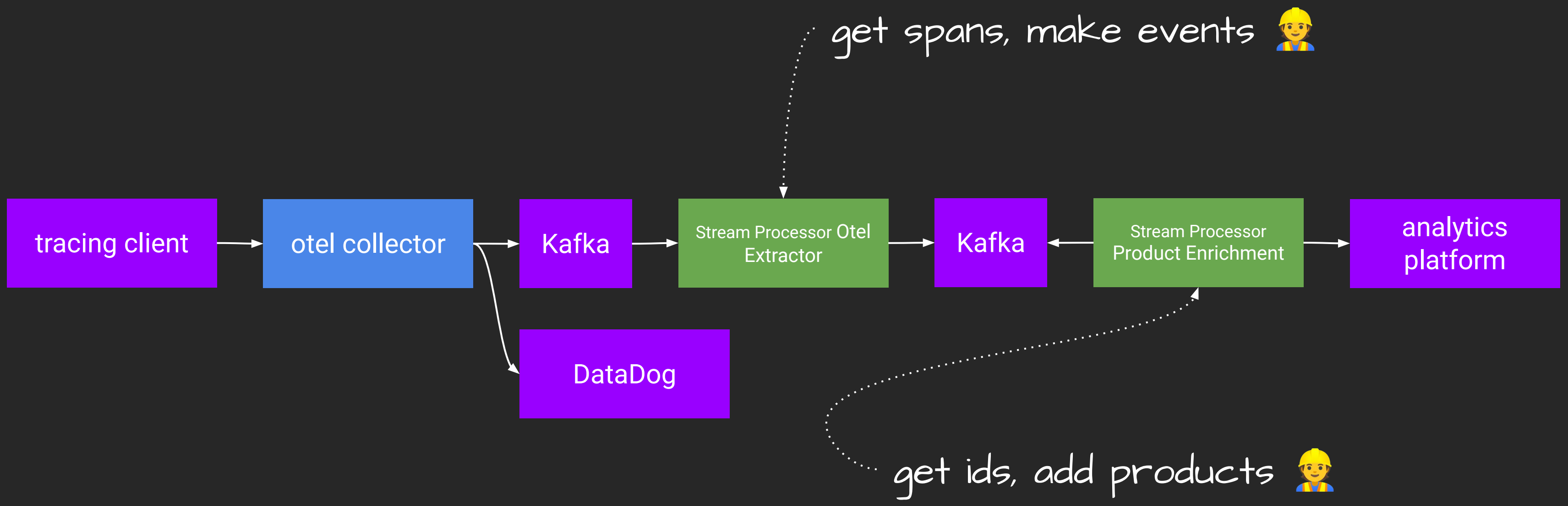
Kafka Streams

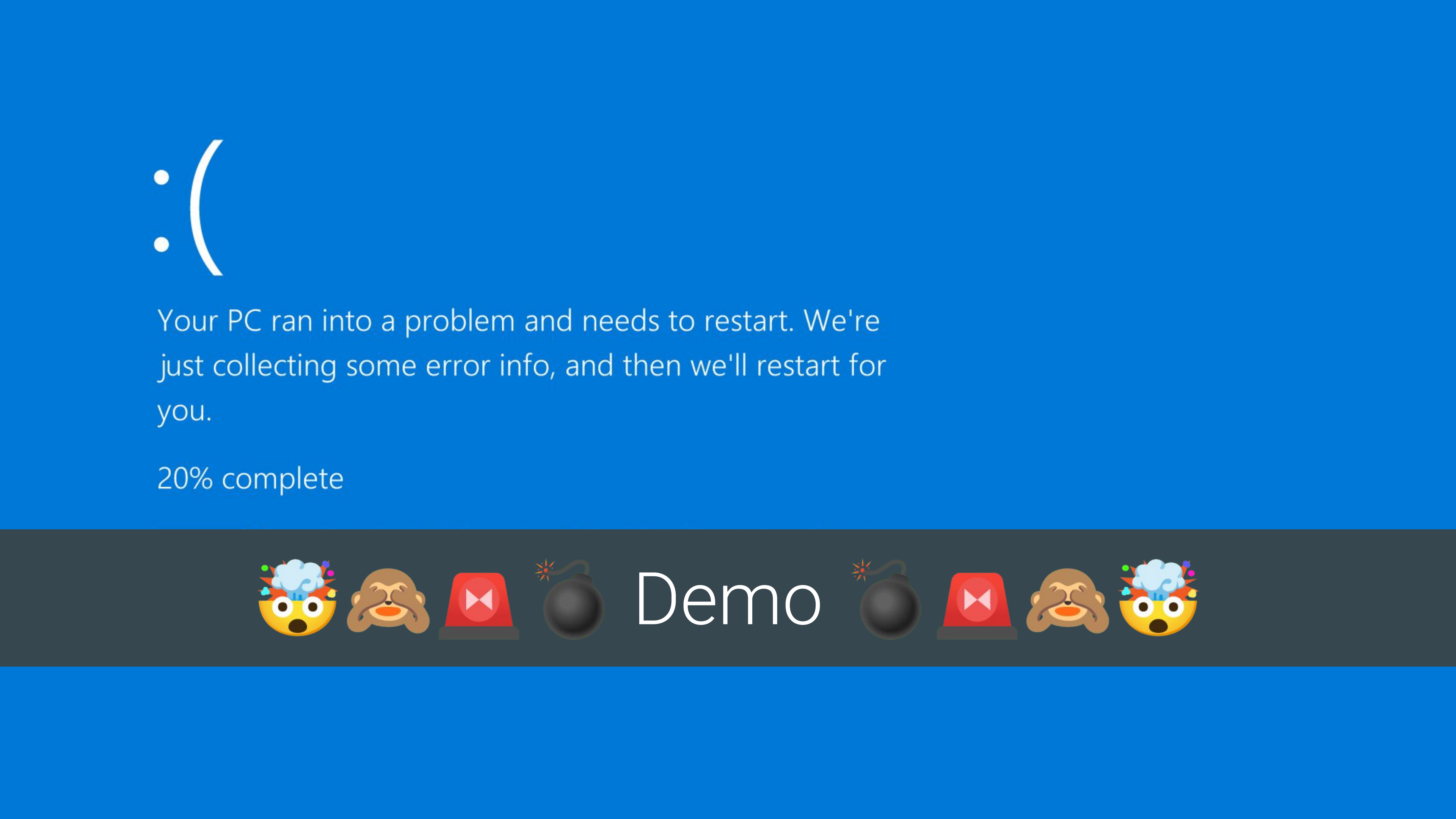
Emoji unicode to image processor



gets grinner face codes only
U+1F600
U+1F60E
U+1F600
U+1F600
U+1F60E
Stream Processor grinner
U+1F600 😀
Stream Processor sunglasses
U+1F60E 😎
partitioned by data key
on-disk k-v database with sunglasses faces only

target architecture





:(

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

20% complete

🤯🙈🚨💣 Demo 💣🚨🙈🤯

https://github.com/Mackie