Data & Privacy, Managed.

White paper: data in Pryv

pryv.com

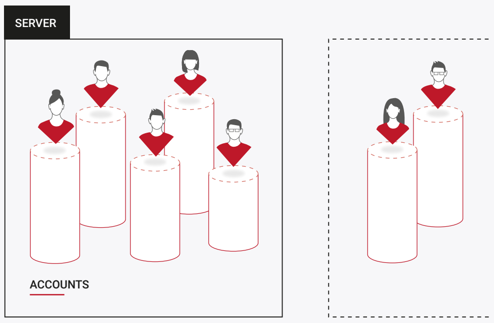
White paper: data in Pryv

This white paper outlines how Pryv manages data and why we designed it this way, hopefully shedding light on Pryv's unique value proposition from the data model perspective.

It's targeted at both technical and non-technical audiences with a basic understanding of data storage and networks.

Privacy at the core

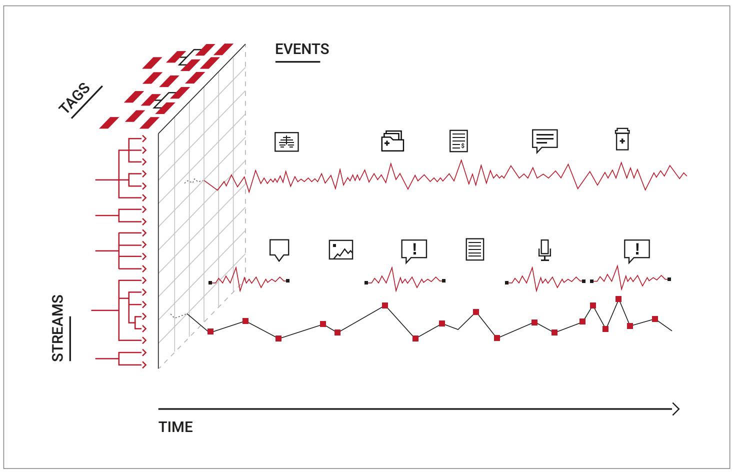
Pryv was designed from day one around privacy and decentralisation. We'll assume you, the reader of this paper, have a pretty good idea why privacy matters. And decentralisation is ultimately necessary to achieve privacy, because centralised architecture implies centralised control, which means people do not truly own their data. So Pryv stores each account's data separately: each account basically has its own database, which can be moved around (on its own dedicated server if needed).



- No big collections holding the information of all users.
- No design headaches with confidentiality.
- A sensible, natural way to store data.

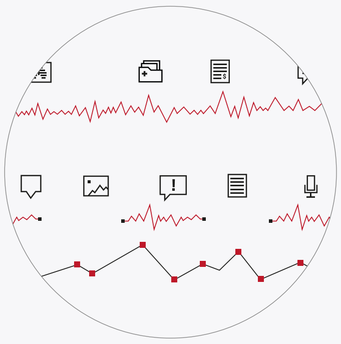
A data structure designed for all

But can people really be said to own their data if they can't understand it? We don't think so. So we strived to design a data model that makes sense to both developers and users. Based on a few simple concepts, with neither too little nor too much abstraction.



Events: the base units of data

To let data be easily understood and exchanged across systems, we provide an open directory of standard types, which we recommend developers use when interoperability is a concern (it rarely isn't). It's worth pointing out that Pryv events are so easily usable and interoperable because we kept the types low-level, holding all necessary contextual information at the organisational level (our next topic). For example: we have no such things as «heart rate» events; instead in Pryv you'll use `frequency/bpm` events classified under the «heart rate» context.



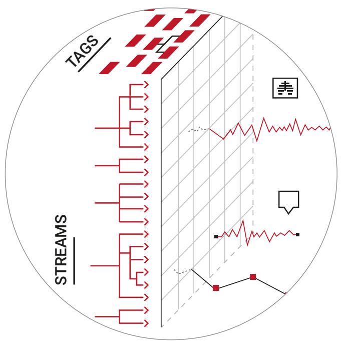
Pryv stores things that happen in time as _events_.

- A blood pressure measurement.
- A note.
- A photo.
- Anything.

To differentiate photos from notes, every event has a _type_.

Tags and streams: ready-to-use organisation

When you record something that happens in time, usually you need more context than just time. Where does the information come from? Why was it recorded? Where shall it be used, and/or by whom? Pryv uses _tags_ and _streams_ to keep track of all that. They're simple, proven constructs to structure data in a free many-to-many fashion (tags) and/or strict one-to-many hierarchy (streams) that covers most – if not all – organisational needs.



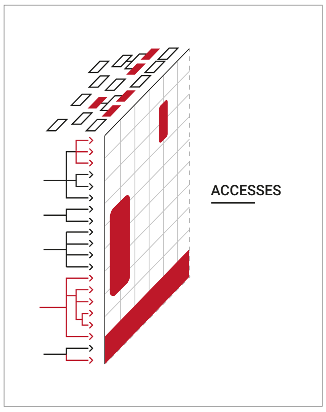
At this point something might become apparent:

Pryv clearly sets itself apart from the many solutions out there which provide unopinionated (fully freely-structured) object storage. Those certainly have their use sometimes, but if we really want to tackle the issues of data ownership and interoperability, we need to balance flexibility with familiarity. So when designing Pryv we paved a few cowpaths – which brought along other benefits for developers, such as a more expressive API.

No-nonsense access management

We have an idea of what data looks like in Pryv (and why), now how do we access it? Every connection to Pryv data is mediated by, well, an _access_.

No surprises here either, we strove to keep it straightforward: account owners connecting via a trusted app use _owner_ (personal) accesses; for other apps they use _delegate_ (app) accesses; to share data with other people they create _shared_ accesses. As expected, the latter two only provide a limited view of the account's data. That view – or subset – is primarily based on context (streams and tags), because that's how we humans handle privacy in most cases. So controlling who sees what is just a natural aspect of properly contextualising data.



Cross-account indexing/searching and aggregation

Pryv keeps every account's data separate from other accounts', so how do you work with data across multiple accounts?

You do it what we consider the Right Way: «client»-side, probably on a middleware service. So for example if you want medical statistics over a number of patients, you'll ask the patients to grant your app/service access, aggregate each patient's data subset into your own database, then compute your stats and/or maybe expose the aggregated data to other apps through an API.

How's that a feature you ask?

We understand it might not look like one today, but we believe it actually is – because there is no way around it if you take privacy seriously. That's just how we think personal data will work in the future. So you might consider building things this way to be a wise investment.

Future-friendly

More and more people realise there's a problem with how most apps and services on today's internet deal with personal data. It's not that they're ill-intentioned, but that they're build on an client-server architecture that's inherently at odds with privacy and proper data ownership.

As a growing number of us work to change the rules of the game, we can see a truly decentralised web coming that fully realises the promise and design of the internet. Pryv is our contribution to that change, and while it still comprises centralised components as of today, its data model is fundamentally ready for tomorrow's web. Building on Pryv means your apps will be, too.

Pryv® SA is an independent, privately-held Swiss company and a pioneer in the development of innovative data and privacy management software for businesses active in highly regulated markets, active since 2012.

Based on years of tests & analysis, we built pryv.io, the Swiss-made personal data management software. Businesses in highly regulated markets, such as the healthcare industry, use Pryv to secure access and complaint management of personal data for the development of scalable personalized products.

Winning with Privacy.