

# In Pod we trust:

towards a transparent  
data economy.

---

Things are totally out of control. We don't know who has our data, what they have and with whom it is shared, for what purpose it is used, nor for how long. Our data is repeatedly collected and processed by governments, private companies, and other organizations often without our consent. Sometimes our data is even used against us.

We founded polypoly as a countermeasure to this; to build a solution that will return control of our data back to us. We will decide who can use our data and for which purpose. We will define the rules. There's no going back.

This paper provides a high-level overview of one part of our technical solution: The **polyPod**. It introduces fundamental aspects, technical decisions we have made – along with our reasoning – as well as some aspects we have not yet addressed. The core objectives of this paper are to set the stage for future white papers which will cover each point in more detail, and to develop a clearer picture of what we are building.

## AUTHORS

**MAIN** *Christian Buggedei, Felix Dahlke* • **CONTRIBUTORS**

*Jacek Bilski, Thorsten Dittmar, Lars Eilebrecht, Lars Hupel, Mira Mezini, Oliver Tigges*

**EDITORS** *Laird Brown, Nils Loeber, Sabine Seymour*





# table of contents

---

I. Fundamental Principles .....	3
II. The polyPod .....	8
III. Trust .....	22
IV. Security and Privacy by Design .....	29
V. Future Outlook .....	35
VI. Summary .....	36
VII. Glossary .....	36
About us .....	38

*We introduce new terminology in italics; we **highlight** important terms covered in the glossary at the end of this paper. For related work we link to relevant external resources in the text.*

## I. Fundamental principles

The current data economy suffers from severe problems resulting in an imbalance of power among market forces.

In parts they culminate in "Surveillance Capitalism", a term coined by Shoshana Zuboff in the book of the same title. The root of these problems is economic and thus can have no purely technological solution. →

➔ But the situation is greatly impacted by how technology is learned and perceived. The misuse of personal data almost always has an economic origin, and mainly affects people without deep technical understanding.

**This is manifested by two symptoms:**

- ➊ There is no human-intuitive grasp of what technology does with our data. What computers do and don't do is increasingly opaque and has very little bearing on how people handle things without computers. Paying with cash in a supermarket is fundamentally different from paying online with a credit card: The former is usually anonymous while the latter creates a surprisingly detailed data trail.
- ➋ The current implementations of technology are based on the notion that data is easily copied and computed, and aim to extract the maximum value from those operations. This sets improper economic incentives and thus rewards privacy-harming behaviour. Building a privacy-friendly and GDPR-compliant system currently means extra costs that generate no immediate value to a company, and can even put them at a competitive disadvantage.

**It follows that in order to properly address these symptoms, a two-pronged approach is necessary:**

- ➊ Technology should extend our natural behaviour to the digital world, but not change it. And where that is not possible, at least the impact of technology must be easier to understand. It is important that everyone has an informed notion of what a machine will do based on their choices.
- ➋ The underlying economy and technical infrastructure needs to reward privacy-friendly behaviour instead of punishing it.

From these insights, we have adopted two key concepts that are fundamental to the polyPod software:

---

## Human-centricity

All user interaction and data models need to be based on the “gut understanding” humans have about their day-to-day interactions and experiences.

This understanding varies from person to person based on a variety of factors, like age, education, experiences, or economic situation. Technology must adapt to these differences.

## Decentralization

Data needs to be decentralized to prevent big data silos and centralized control. It should be computed at the edge of the network, on the end users’ devices. This requires a radically different approach to operations and problem solving, since it incentivizes privacy-friendly behaviour and punishes privacy-unfriendly behaviour. The current data economy suffers from severe problems resulting in an imbalance of power among market forces. In parts they culminate in “Surveillance Capitalism”, a term coined by Shoshana Zuboff in the book of the same title. The root of these problems is economic and thus can have no purely technological solution. But the situation is greatly impacted by how technology is learned and perceived.

The misuse of personal data almost always has an economic origin, and mainly affects people without deep technical understanding.



## Impact

### ... on individuals

Individuals gain more control through a better understanding of digital processes as well as by actual Data Ownership. This enables better services without the obligation to trade data for those services. Having all data in a structured format in one place empowers individuals to make better sense of it. They also gain more data protection, better data-based services, and the option to directly participate in the monetization of their data if they choose. Because the algorithms operating on the individuals' data are executed on their polyPod, individuals gain detailed insights into who uses their data to what end.



### ... on enterprise

Companies switching to the polypoly environment can profit from lower costs since they will no longer have to store and compute large amounts of data. At the same time access to higher quality customer data enables enterprise to offer better services, thus being more competitive while protecting user privacy. Seamless integration of the polyPod data lowers friction.

A decentralized system where data is handled only at the networks' edge has several benefits: Companies can lower their data centre costs; their liability risks; access higher quality data; and vastly improve the remaining data handling processes. Since they will not be responsible for a large treasure trove of data anymore, data breaches and the resulting negative publicity and possible fines and damages will be drastically reduced. →



## ... on the government

It is much clearer which local laws apply: Computations and data storage take place on the user's device. As a result legal vacuums do not exist in this context and taxes, for example, are always due in the user's country. In addition, official tasks can efficiently be carried out digitally without dependence on third parties.

*For a deeper dive into this topic, see our [Economical Whitepaper: Digitale Ökosysteme \(German\)](#).*



## ... on the community

A digital ecosystem rewarding privacy-friendly behaviour creates the opportunity for a seismic shift in the economy. Currently – due to the market concentration of US technology giants – other parts of the world have a difficult time participating in that wealth-generation. By ensuring that the data compensation is kept on the side of individuals, any wealth creation can be maintained within localities too, creating a more level – and more fair – global playing field.


## II. The polyPod

**The polyPod is a container hosting an individual's personal data.** It provides infrastructure for services working with that data, and governs these interactions.

*In this section, we explain the overall architecture of the polyPod.*

## Its basic functionality can be summarized as follows:

- ❶ It **stores** personal data.
- ❷ It **provides** an execution environment for **Features**, which run algorithms on the stored data and interact with the owner of that data (**Pod Owner** from here on).
- ❸ It **manages** the individual's **Identity**, and which pieces of data belong to what **Facet**.
- ❹ It **manages** data access rights for Features.
- ❺ It **handles** communication between Pod Instances, Identity Facets, and the outside world.
- ❻ It **comes with** a few pre-installed Features that enable individuals to import, browse and interact with their personal data.



The term **Pod** is also used by **Solid**, but the two systems are only loosely related and differ in fundamental ways which we will describe in a separate white paper.

## Storing personal data: Data model based on user intuition

Humans rarely think of their personal data in terms of tables or files. Instead, we remember meeting people, friends and family, circumstances and life events. As such, we think of personal data as discrete but interconnected data points, where connections are usually coupled with a certain sense of identity.

Another core idea – not very well in line with how data is typically handled by software – is that all personal data belongs to the user. No piece of data belongs to a Feature. That means that while Features can store data they do not have exclusive access to it. The polyPod has a single pool of data all Features could potentially access.

### Linked data

To achieve interoperability between Features, all of the data stored in the polyPod is represented as **Linked Data**. **Example:** Jane installed two different Features: A Facebook importer, and a LinkedIn importer. Both Features downloaded her data from their respective platforms and stored it in her polyPod. While these are two different accounts on two different platforms, Jane can now link Facebook friends and LinkedIn contacts referring to the same person. Technically, that person will then be persisted as a single resource with a few properties, for example their phone number. Defining this will add an additional triple to the data pool which has the form subject-predicate-object.

**In the Linked Data paradigm, each piece of information is represented as a triple with subject, predicate and object:**

- 1 **The subject** is the resource being described.
- 2 **The predicate** defines the relationship between the subject and the object.
- 3 **The object** may be another resource or a literal value.

Storing data in such triples enables us to describe things with as much – or as little – detail as necessary. This enables Features to provide a wide range of functionality: An address book Feature could show a person in an aggregated view containing information from all sources. A chat Feature could show all messages exchanged with that person, regardless of which system those messages were sent on. A calendar Feature could remind about their birthday, a photo gallery Feature could show pictures with that person, and a genealogy tree Feature could show the family relationships of that person.

## Schemas

**Such great flexibility comes with a significant problem: Tools reading and processing such data need to understand those predicates.**

In the example above, a person's name can be described using different predicates such as "last name", "family name" or "surname". A human would understand that those are all synonyms, and generally make sense of a situation where the same information is presented in a different way.

Features may store whatever triples they deem to work best for them and Pod Owners. And to make things more complex: Neither do we plan to define new schemas for Features.

We realize that this might lead to a proliferation of schemas, and that Features may define custom schemas that cannot be readily understood by other Features. This is a trade-off we are willing to make – we trust Feature developers and Pod Owners to make their own choices.

We expect to see a lot of new schemas, however: Whole industries might need to develop schemas for storing data they work with, such as insurance contracts, travel offers, car maintenance requirements, and so on and so forth. We encourage developers to extend existing schemas in a non-proprietary way, i.e. accompanied by an open specification and designed with interoperability in mind.

Ideally, whoever builds a Feature around a specific set of data should set the precedent and standards for storing that data.

There are already well-known and widely-used schemas for Linked Data such as [schema.org](http://schema.org), [Dublin Core](http://dublincore.org/), [FOAF](http://foaf.org/) and others.

Algorithms, however, interpret data based on strict rules. We remedy this problem by encouraging the use of **schemas**. There are already well-known and widely-used schemas for Linked Data such as [schema.org](http://schema.org), [Dublin Core](http://dublincore.org/), [FOAF](http://foaf.org/) and others.

They provide common, comprehensive dictionaries all polyPod Features can use to be interoperable on many, but not all concepts from the real world. It is important to note, however, that the use of existing schemas is encouraged but not mandated.

## Execution environment for features

**Since the polyPod itself is primarily a container, all the noteworthy functionality is provided by Features.**

Features can do whatever the Pod Owner wants: Visualizing, editing, or deleting existing data, importing new data, providing or even leasing access to the data, communicating with other individuals ... anything really. They are crucial to the success of this project and we remain confident that Feature developers will innovate beyond what we – or any one organization – can provide.

On the other hand, Pod-Owners will only install Features that they find useful. Nobody can dictate how they should make use of the data they own.

This approach is the opposite of the trend we currently observe where more and more data is being kept “in the cloud”.

Companies keep their users data in their own systems and use it as they see fit. Since monopolies control this data, it is primarily being used for their benefit; certainly not for the benefit of users. We aim to solve these detrimental dynamics by reverting the direction: In the polyPod, data does not flow to companies, but algorithms provided by companies need to be sent to the Pod Owners, to be executed on their systems, transparently, and according to their rules.

We call those algorithms Features. From an end user perspective, the polyPod is a platform, and Features are apps that can be installed on that platform, but we deliberately do not use the term “app” to clearly differentiate from the current connotations of the term “app”, based on how smartphone apps typically operate today: They are usually just local user interfaces, while the data and functionality lives “in the cloud”, outside the user’s control. Features on the other hand mostly use data stored locally on the polyPod, performing computations directly on the device. →

Features can do whatever the Pod Owner wants:



→ We have already summarised the benefits of this approach above. But to recap and become more concrete. We see tremendous advantages in our approach over the status quo:

❶ **Individuals will gain full control over their data.** They can understand and control what data Features have access to, and what they do with it.

❷ **Because the data Features can access will be coming from multiple sources** – and will be managed and annotated by the Pod Owner – Feature developers will be able to work with more robust, precise, and meaningful data sets than they could on a platform where a single company collects all the data, and decides what other parties can access it.

❸ **Pod Owners share their data with other individuals** or organizations deliberately rather than accidentally: They decide what they want to share, with whom, for how long, and at what price. This dynamic ensures that individuals profit from their data, by gaining the functionality they want, by supporting the causes they want to support, or by directly selling specific insights. The choice is theirs.

❹ **While larger companies generally profit from owning a lot of user data, it also comes with inherent complexity and risk:** They have to worry about acquiring, storing and protecting that data, complying with *data protection laws like GDPR* and securing their systems against the bad actors that a valuable pool of data inherently attracts.

❺ **For smaller companies and startups** that are unable to shoulder the burden of acquiring and storing a lake of user data, the polyPod is a blessing: Companies and organizations of any size or means can compete on the same playing field.



## Write once, run everywhere

**While we aim to bring the polyPod to a wide range of platforms we follow a write once, run anywhere approach when it comes to Features.**

Once written, a Feature will be able to run on any device for which the polyPod is available. Naturally, not all Features are a good fit for every platform: A Feature for sharing pictures might not be a good fit for a polyPod running on a smart watch. To account for these differences, Features may decide to support only devices with certain capabilities, or offer different functionality depending on device capabilities.

Specific capabilities aside, the polyPod offers a uniform API for Features that allows them to access the data, communicate with the outside world, or interact with the user.

---

A Feature will be able to run on any device for which the polyPod is available.



The Feature execution environment interprets ECMAScript (commonly referred to as JavaScript). Due to the immense popularity of the web, and JavaScript being the only programming language directly supported on that platform, Feature developers have access to the vast JavaScript ecosystem, allowing them to work with a host of existing libraries and frameworks, and to write code in any language compiling directly to **JavaScript** (TypeScript, Scala.js, ClojureScript, etc) or **asm.js/WebAssembly** (C, C++, Rust, etc.).

For security reasons, we sandbox the JavaScript runtime used in all polyPod implementations. For example there will be no direct support for network calls, but only through special polyPod APIs, which provide transparency over what a Feature is doing, and adhere to the Pod Owner's decisions on what a Feature should be able to do. Due to the removal of common APIs like `XMLHttpRequest` and capabilities like string evaluation, existing JavaScript code may not work in the polyPod without modifications. But we believe that this is the right tradeoff to make. →

→ In addition to this technical abstraction between a Feature and the device on which it runs, we also plan to abstract the interaction between Features and users, e.g., graphical user interfaces and visualizations. This further enables us to adjust the user interface to the knowledge, needs and wishes of the end user: A teenager spending a lot of time in social media has different knowledge and needs when it comes to their personal data than an elderly person who largely avoids software. If our approach works, we can meet our goal of enabling Feature developers to write a Feature only once, for all devices and target audiences.

To realise this, we are developing a declarative language called **polyLook** that allows developers to describe how data should be visualized, and what interactions are needed. The polyPod will then render the data according to the capabilities of the device and the Pod Owner's profile and preferences.

That means that feature developers will have only limited means to make the user interface look a certain way, they can only describe their intentions. The polyPod interprets those intentions and acts on it based on the needs and preferences of the end user.

x

- o
- o

Features do not have the ability to directly communicate with any device or network.

---

## Communication restrictions

As mentioned above, Features do not have the ability to directly communicate with any device or network. Those operations must be requested through the polyPod's API, and will then be performed by the polyPod on behalf of the Feature. Network calls are the biggest privacy risk for Pod Owners, therefore many restrictions apply: Features can be restricted to only contact specific domains or IP addresses defined in their manifest; explicitly allowed by the Pod Owner, or to only send data (or computations based on data) explicitly allowed by the Pod Owner.

When communicating with other applications running on the same device, the polyPod API offers this as well. It has similar restrictions to support use cases such as a Feature offering a CalDAV service, so Pod Owners can continue to use their favourite calendar app.

## Feature depots

### Pod Owners can download and install Features from Feature Depots.

If a Feature is similar to an app, a **Feature Depot** is similar to an “app store”. However, we again decided to introduce new terminology to differentiate from what people commonly associate with the term app store:

The necessity to comply with all the rules a single powerful company defines, to protect their business more than their users. To avoid that dynamic, any organisation or individual can provide a Feature Depot and individuals can decide to trust it or not.

There will not be any license fees, mandatory certifications or other requirements for Feature Depots.

All they need to do is implement the interface that allows the polyPod to interact with it, and to convince Pod Owners to use their Feature Depot.

If a Feature is similar to an app, a Feature Depot is similar to an “app store”.

### Each Depot can set their own rules.

For example, defining review criteria or charging Pod Owners for Features (similar to buying an app on an app store). To kick things off, polypoly will operate a Feature Depot, but we hope to see other people and organisations set up their own Depots.

Because the polyPod’s **trust model** applies to Feature Depots as well, an organisation with no – or even a bad reputation – will first need to build trust in order for a significant amount of individuals to use their Depot.

Similarly, if the individuals lose trust in polypoly, its Feature Depot will lose traction. Just like the human concept of trust, trust in the polyPod’s model takes long to build and can quickly evaporate based on an organisation’s or individual’s actions.

## Identity

**A crucial concept that we build upon are Identity Facets. Every person has as many Facets of their online Identity as they do IRL.**

For example: A single person can be a family member, an employee, a bowling club member, a Greenpeace activist; all at the same time. Another term commonly used for this is “roles”, but we call them Facets. Even though different Facets belong to a single person, they can be quite separate from each other. An example: A person who is both an activist and a member of a bowling club does not need their fellow activists to know that they are a member of a bowling club, and vice versa. Each Identity Facet of a person contains different data, while some data may be shared across Facets.



For example: The same private email address used to communicate with a person’s family can also be used for communication with their bowling club. Conversely, people are less likely to use their company email address to communicate with anyone other than co-workers and business contacts.

Facets are also the basis for all external communication. An individual’s entire Identity is never exposed, nor is there a concept of something like a “Super Facet” that includes all their data. Facets can have different trust levels, ensuring that the more private and accurate data is only accessible to highly trusted parties. Some form of communication may however require certain type of Facets – for example; a Facet covering an individual’s identity from the government’s perspective may be required for interacting with their local tax authorities.

Every person has as many  
Facets of their online Identity  
as they do IRL.

## Data access rights

**Due to its design the poly-Pod aggregates data from a wide range of data sources: There will be more data about a single person than even the most powerful companies and governments in the world can collect.**

Therefore, Features are restricted to which data they can access. Technologically, one hurdle is that no Feature is likely to understand all **schemas** that are being used to store data.

But more importantly, Pod Owners are able to restrict Features, and define rules and conditions for governing access to their data through Identity Facets. An individual may even use the same Feature with different Facets, e.g., to represent two different accounts set up on the same social network. →

---

An individual may even use the same Feature with different Facets.



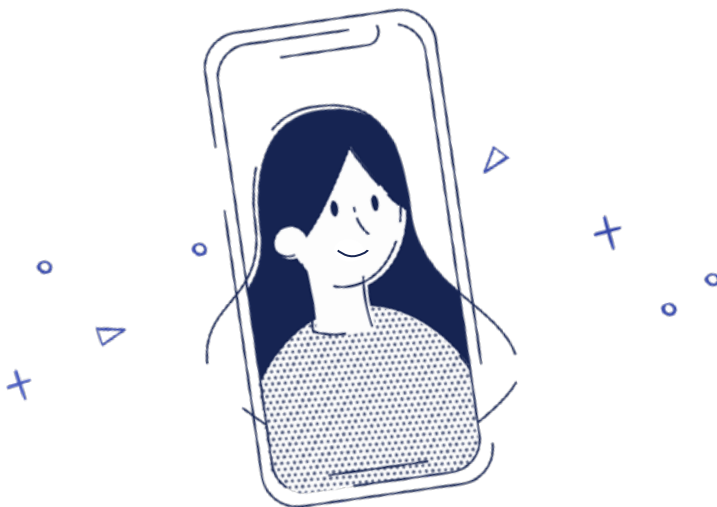
→ While these access restrictions are controlled by the user and supported by the polyPod's trust model, we take additional steps to secure the Pod Owner's data based on three principles: review, enforce and audit.

**① Features contain a manifest file that declares the permissions for which they ask.** This manifest file is scrupulously reviewed by the operators of the Feature Depot; certainly for the one operated by polypoly. A Feature that does not need network access to function should not declare it in its manifest.

**② The polyPod enforces – through the API – that no Feature exceeds its declared permissions.** Pod Owners may restrict this access even further, however, we are aiming to provide reasonable, secure-by-default manifest files.

**③ Optionally, the source code of the Feature itself may be audited by the operators of the Feature Depot.** This requires implementers to provide their source code, because a Feature's JavaScript code may be obfuscated or translated from a different source language. The polypoly Feature Depot will require audits for Features that request particularly sensitive permissions, e.g., access to sensitive health data.

The immediate consequence is that Pod Owners do not bear the burden of ensuring their own security and privacy manually. If a Feature attempts to retrieve data that is restricted the polyPod would deliver an empty result set, even if that data actually existed. Consequently, from a privacy perspective, a Feature cannot distinguish between data not existing, or data not being accessible to it.



## Network communication:

### One Pod

**The polyPod is a piece of software individuals can install on their devices.**

A Pod, on the other hand, is a logical entity: A collection of all the data of that individual, which is always a single, natural person. All the Identity Facets of that individual exist in the same Pod. Separating Facets into different polyPods would lead to a range of problems:

- ❶ It would be unnatural. After all, all those different Identity Facets refer to “us”. We cannot clearly separate one Facet from another in our heads.
- ❷ The tension between what feels natural and how the system is structured would lead to a plethora of problems we cannot predict. Our goal is to make our solution as natural as possible so it is easy for everyone to use and understand.
- ❸ Because Facets quite often overlap, there would also be a lot of data duplication and data synchronization challenges.

## Multiple instances

A Pod is a logical entity. The tangible parts are called Instances. An instance of a Pod exists as soon as an individual installs the polyPod on one of their devices. A single Pod Owner can have as many Instances as they want, and they even need several in order to use the polyPod on multiple devices.

Logically, the Pod is the sum of all those Instances. As opposed to Pods, Instances may hold different sets of Identity Facets: An individual might decide to have their professional Facet present only on company issued devices, but not on their private devices. The Instances will synchronize data between each other, making the same data available on multiple Instances, and mirroring all changes to data sets, such as adding, removing, and modifying data entries.

## Instance to instance communication

Synchronization between Instances will be supported through a *peer-to-peer system*. There will be no central servers; the Instances will discover each other and synchronize automatically. The Instances will also default to solving potential synchronization conflicts automatically without bothering the Pod Owners with the technical details of resolving conflicts between data entries.

## Facet to outside world communication

As described above, Features have local access to the data from an Identity Facet; they are able to communicate with the outside world. In order to ensure security and privacy, we introduce additional concepts and restrictions. For instance: The polyPod does not allow any unencrypted connections and takes steps to anonymize communication to prevent tracking of Pod Owners. We elaborate more on this in the **security** section.

## Facet-to-facet communication

Apart from Features communicating with external systems, they can also contact other individual's Identity Facets. Accordingly, Features can offer functionality such as messaging or data sharing between Pod Owners. This protocol is available via the Features API of the polyPod. We call this Facet-to-Facet communication since Pod Owners will be the only persons knowing about the existence of a given Pod, and the Facets within it. They can only reveal a particular Facet on a case by case basis. This approximates the offline world, where an individual can introduce themselves to someone as an employee of a certain company without revealing all their other roles, such as club member. We elaborate on this in the security section as well.

### **III. Trust**

**In the context of this paper Trust addresses the question of why people should put their data – and thus trust – into the polyPod.**

## The answer has three parts:

- ❶ The core concepts for the polyPod and their implementation contain a powerful, built-in trust model.
- ❷ The polyPod is based on an open platform where third parties can provide all the important components. Nobody needs to trust polypoly as an organisation.
- ❸ polypoly works to create transparency for all participants by publishing relevant information.

*It is important to know that Trust is not a fixed value but one that fluctuates. It can be gained slowly, lost easily, and is highly dependent on a variety of factors.*

## Trust model

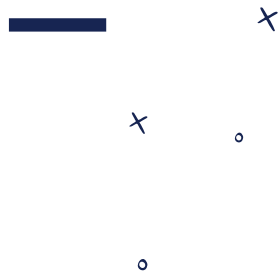
**It is important that people trust the system as a whole. This is crucial for a data economy of any kind to flourish.**

It will be up to the Pod Owners to decide if they trust something or not and what the implications of that decision are.

Usually trust models focus on identities. When people are, for example, communicating with a peer over a network, they want to ensure that their communication partner is really who they claim to be. A common approach for identity verification is certificates issued by a trusted authority. An alternative is a transitive trust model: The amount of trusted peers plus their degree of trust in some other peer determine the level of trust. In those cases, there is no need for a central trust anchor.

polypoly will not have a mandated global trust authority within its infrastructure. Instead we will use a transitive trust model. However, this trust model is not only about verifying a peer's identity. In real life – if a person is about to share personal data with someone – the question is not only if this partner is really who they claim to be, but also how they assess reliability, honesty, responsibility, and competency.

If a friend suggests you some company as trustworthy, your assessment of your friend's reliability and competence in the judgement of this company will be considered in your verification process. But trust is not only applicable to people and companies. The same ideas are used around other aspects. For example, the fact that polypoly will be hosting a Feature Depot and will be carefully reviewing and approving Features doesn't automatically mean that the Depot and Features should be trusted.



## Trust must be earned.



### Once people start using and rating the Features, the trust level might go up.

But if the polypoly vetting process allows others to do harm to a persons polyPod and data, people may stop trusting it. Another potential problem is when a Feature that was behaving properly suddenly starts doing malicious things. In that case Pod Owners can quickly stop trusting it and others would see that something is wrong.

But it will be up to the Pod Owners to decide if they trust something or not and what the implications of that decision are. Pod Owners can still decide to install an untrusted Feature or connect to an untrusted Depot if they want to.

**polypoly will design and implement a multi-dimensional, transitive trust model where identity is only one (important) aspect of the verification process among many social factors.**

## Trust patterns

**While a person usually trusts their parents experience with their bank – and thus do trust the bank too – they might find that their parents food preferences a little bit ridiculous (For example someone’s mother cannot fathom that they don’t like mushrooms).**

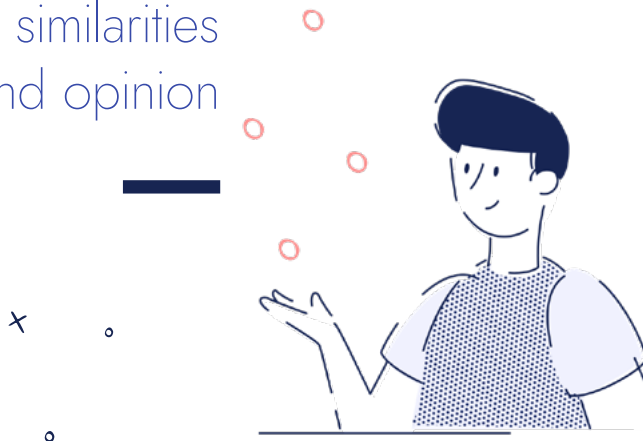
So they choose to not trust their parents food recommendations. (As they find mushrooms being sneaked onto their plate). **That is a Trust Pattern:** Parental recommendations for banking information are transitive while food recommendations are not.

We use those Trust Patterns to denote areas of expertise (understanding how something works), experience (knowledge gained through time), similarities in taste (which food is good) and opinion (should companies process synthetic fibres?).

There are various sources for these Trust Patterns: Family, friends, organizations, companies, and even governments. Each of these come with their own areas of perceived expertise and qualifications, and each can – and often do – include patterns from other entities. The Web of Trust as implemented in PGP (***Pretty Good Privacy***) is a good and basic model of this.

Pod Owners can state explicitly which Trust Patterns and from which entity they want to use, which in turn may incorporate other patterns. They can also state exceptions or have patterns compound (i.e.: if two patterns agree that something is trustworthy, the resulting trust level rises).

We use those Trust Patterns to denote areas of expertise, experience, similarities in taste and opinion



## Open platform

**The polyPod is based on an open platform. This means that there is no central entity operating as the sole provider of any component. Pod Owners can pick the components from the vendors they trust.**

polypoly provides the initial version of the polyPod software, as well as a Feature Depot and some core Features. polypoly will not restrict others from building compatible products.

Over time, hopefully, others will build their own implementations of the polyPod or operate Depots with differing characteristics. To support this open ecosystem we will provide development and compatibility kits. Those will be sets of tests that anyone will be able to run against their own polyPod or Depot implementations to verify if they could interact with other implementations correctly.



## Open standards

polypoly will use open standards to ensure that our applications and specifications can be used and implemented on a royalty-free basis.

## Open source

The entire software stack from what is installed on end user devices to the software that runs on polypoly backends (the Feature Depot for example) will be open source.

Only open-source licenses that are compliant with the definition of the **Open Source Initiative** will be used.

At any time everyone will be able to review the code, the issues reported, build logs, tests, etc. They will also be able to send patches with bug fixes or improvements. Everyone will be able to fork the code and build their own polyPod with added or removed functionality.

## Transparency

**Our aim is that everyone can trust the software and applications without requiring blind trust in its maintainers. This requires open and transparent processes regarding software development and information on how we deal with security and data protection issues.**

### Software development

Open and transparent development processes for all software components will be implemented. Details about software bugs and issues will be publicly accessible, or details will be published accordingly.

### Information security

Security vulnerabilities – or even potential vulnerabilities – cannot always be discussed publicly as that would put users at risk of being exploited and having their data stolen. We will implement a responsible disclosure policy for handling vulnerability reports, and work closely with security researchers. Information about vulnerabilities and privacy-impacting issues will be published once it is safe to do – when they are publicly known anyway – or when discussing it publicly is the best course of action.

## Data protection

Even with the best security measures and controls in place, security incidents can still happen. Once it is safe to report on these incidents, all details will be published including information about root cause. And details on how we will make sure the same issue won't happen again. Incidents may also affect third-party polyPod Features: For example, consider a Feature from a trusted company that started sending data to an unauthorised destination without any notification or consent from the Pod Owner.

Consequences arising from this behaviour may be a public bulletin, and depending on severity, curtailing the permissions of the Feature, lowering the trust in that Feature, or in extreme cases, removing the Feature from the Depot. Similar consequences would arise from Features exploiting security vulnerabilities in the polyPod application. polypoly will be as open as possible with the Feature Depot. The point is not to gatekeep access to it, but to inform the Pod Owners about what a Feature does and how trustworthy it is. We will also use our public data privacy database polyPedia to publish information about such incidents.

## **IV. Security and privacy by design**

**Because this whole project is about everyone owning their own data and keeping it on their personal devices, we need to ensure that it remains safe.**

To that end, security has to be considered from the get go. The first step is the acknowledgment that people have different security and privacy needs, and accept different inconveniences supporting the required level of security and privacy. →

→ We address this problem by building a system for the average person. It is secure enough to protect their privacy and as easy to use as a door lock.

The needs of journalists, lawyers, politicians, etc. are significantly different. In a later course of the project there will be a specification for a high security polyPod, but that is not the focus of this paper. In the real world, security does not only result from defensive measures. Another important part is the clear codification of law and the possibility for the consumer to enforce this law. This part will only be dealt with in this paper where it is technically relevant.

## Identity & identifiers

**While the notion of identity is largely a social one, there is also a need to define the technical concept of *identifiers*.**

Each Identity Facet, Instance, or Feature Depot will have one, unique identifier. From a security perspective, the natural choice for an identifier is a public/private key pair. This provides a few desired properties for public/private keys, such as:

- 1 They can be used for authentication purposes.
- 2 They can be used for establishing secure, encrypted connections.
- 3 (Raw) public keys have no metadata attached to them, so they do not reveal any information about its owner.
- 4 They can be generated in a decentralized way, i.e., no registration required.
- 5 They can be used to encrypt and decrypt the polyPod data.

## Identity security

**Identity Facets need as much security and protection as the Pod they are contained in. Whereas Pods keep personal data, Identity Facets represent aspects of a person to the outside world.**

As such, making sure that there is no possibility of correlating two distinct Facets is a fundamental part of securing Identity Facets. To stick with the example above: The Facet as a Greenpeace activist may require more protection than being a member of a bowling club. Some might wish to keep their activism secret from employers, colleagues, or family.

Therefore, a new identifier will be created for every Facet. Moreover, those identifiers will be created in such a way that it will be impossible to find out that two Facets belong to the same person.

**Consequently, it will not be possible to deduce any correlation between:**

- ❶ A Pod and an Identity Facet.
- ❷ Two Instances belonging to the same person.
- ❸ Two Identity Facets belonging to the same person.

Another safety option is that Pod Owners can assign a separate password to any Identity Facet. This allows them to unlock the Pod so that only the data for that particular Identity Facet is visible. This might happen if they were forced to log into the Pod by a third party, thus providing plausible deniability.

## polyPod Instance Security

The polyPod is the heart of the polypoly environment. It hosts all of a persons data and that needs to be very well protected. It needs to be secure and the necessary functionality to ensure privacy needs to be built-in. Everything that polyPod instances store on disk or other storage media is encrypted and only the polyPod Owner is able to decrypt it. Even when attackers get hold of the device, the data will still be protected.

## Secure backup and recovery

As polyPod Instances are the repository for all personal data, polyPods will be extremely valuable for their owners. Password protection ensures that unauthorized persons will not be able to access the data. A recovery mechanism safeguards the Pod Owner can regain access. For example, in the case of a forgotten password.

Additionally, Pod Owners can back-up their entire Pod and keep it somewhere safe. Due to the fact that most Pod Owners will probably have multiple Instances, they will have a basic backup already.

To achieve this there should be social as well as technical solutions. For example, Pod Owners will be able to get access to their Pods back if enough people from their social circle confirm their identity. Similarly, there should be distributed backups: This means that all the data from a Pod will be distributed across other Pods in such a way that it will be possible to recover the Pod and all the data. At the same time this distributed data will be inaccessible to others.

The polyPod is the heart of the polypoly environment.

## Networking security

**All communication is encrypted. That should be the global standard for any network communication.**

As well, Pod Owners should have a right to be anonymous and prevent others from learning about them. To ensure this, polyPods will form a peer-to-peer network that enables Pod Owners to hide their communication, and even themselves, as much as possible. For example, any connection between Pod Instances will be anonymized. It will be impossible to find out that any two Identity Facets belong to the same Pod or person, just by looking at the network traffic. It is also impossible to find out if a Facet or an Instance using a given identifier exists.

Connecting people and Instances, etc., to each other will always require a user intervention and an out-of-band confirmation. Other exceptions, such as downloading a new Feature update from a Feature Depot will require user confirmation or consent.

## Economic perspective on data security

In the long run the data ecosystem becomes more robust to security incidents: In the polypoly environment, the number of large databases with millions of user profiles will be greatly diminished. Instead, personal data will only be stored on the private systems of each user. This is better because the economic incentives to break into these systems are significantly lower.

Breaking into a computer system takes time and effort. This effort is rewarded if there is a wealth of data behind that system, compromising millions of accounts. Having to break into every person's computer individually takes more time and effort yielding only small rewards. In other words, if hacking and/or abuse of personal data becomes too expensive, too risky, or just too much work, it will come to an end. We see good examples in the real world, such as the door lock on our apartments. It's a relatively simple lock and not really a hindrance for professional burglars. Yet most of us are spared from having our apartments broken into.

## Simplification

**It is important to note that strong and multi-layered information security will always be more complicated and will require more mental effort to use.**

Thus the polyPod's user-friendly UX gives it the equivalent of a sturdy-but-simple door lock. Still, each of us has their own. That is to say, it is decentralized.

In order to protect the "average" user from attacks by companies or organized criminal hackers, all the necessary decentralized technologies have already been invented and available for years.

**But they are often not used because they are:**

- 1 Too complex to use for non tech people.
- 2 Not embedded in a meaningful ecosystem.

Our premise is that if we come up with a solid, easy-to-use and decentralized system, the first part of the problem will be solved. If we then find a way that is comfortable for the user and provides the economy with the relevant information (cheap, trustworthy, and without harming the privacy of the user), then it will become a game changer.

Besides decentralization, simplification is a key aspect. The conventional approach to enhanced end-user security relies on empowerment and education. It is almost always meant that people should learn how the digital world works. However, even experts have difficulty understanding the security implications of complex systems. We believe that systems should foster an intuitive understanding of the consequences of digital actions. If that understanding is absent, then the fault lies with the software systems, not with the users.

## At-risk persons

Under no circumstances do we want to gloss over the fact that there are particularly vulnerable groups of people, or that even states can be violent, be it their own or a foreign one. The technological adversaries for those people either do not have to worry about that cost-benefit calculation, or simply do not care. For such people, additional security is vital. polypoly will eventually release a special edition of the polyPod that can provide that extra security.

This will come at the expense of usability otherwise it would of course be part of the general polyPod.

## V. Future outlook

As this is only the beginning of the polyPod story there are a number of concepts and functions that we haven't fully addressed.

These will be discussed in future whitepapers and include:

- ① Branded and unbranded Features.
- ② Temporary Features with a subset of permissions.
- ③ AI-based Features.
- ④ Federated machine learning and federated Big Data queries to enable large scale privacy-conscious research across millions of Pods.
- ⑤ A thorough explanation of our data privacy and trust database polyPedia.
- ⑥ High security polyPods.

## VI. Summary

In this paper we provided a high-level overview of how the poly-Pod works and why it is designed the way it is. We only scratched the surface of this project here.

This is a big and profound undertaking. The problems are complex and it will take us some time to solve them and present all the solutions. We will do so by continuously releasing additional white papers as we progress.

## VII. Glossary

**Data Ownership** The concept that all data relating to an individual's life belong to that individual, and need to be under their full control, primarily benefiting that individual – just like their physical possessions. This is in contrast to the status quo, where companies tend to collect the data of many individual's in an obscure way and use it for their own benefit.

**Feature Depot** A piece of software Pod Owners can install in the polyPod to visualize or otherwise work with their data.

**Feature Depot** A repository for Features; technically comparable to the Google Play Store, npm, Maven Central, and other software repositories or app stores.

**Identity** The sum of an individual's Identity Facets in the polyPod context, when not capitalised the general concept of who a person is.

**Identity Facet** A subset of an individual's data, representing an individual's different roles in society and their willingness to share data as well as their needs connected to those roles.

**Instance** The installation of a polyPod on a specific device.

### Linked Data

Interlinked, structured data based on Web technologies, see [linkeddata.org](http://linkeddata.org) →

## VII. Glossary

**Pod** The collection of identities and data of a Pod Owner across all installed instances of one Pod Owner.

**Pod Owner** An individual who uses the polyPod, and consequently owns a Pod.

**polyPod** A piece of software individuals can install on their devices to collect and organize their data and install Features that operate on it; software that enables Data Ownership.

**Schema** A specification on how to store and parse a set of structured data. The schemas are a set of 'types', each associated with a set of properties. The types are arranged in a hierarchy.

**Solid** As per <https://solid.mit.edu/>: Solid (derived from "social linked data") is a proposed set of conventions and tools for building decentralized social applications based on Linked Data principles. Solid is modular and extensible and it relies as much as possible on existing **W3C** standards and protocols. The polypoly ecosystem will be in many ways compatible with Solid.



## Join us in our mission!

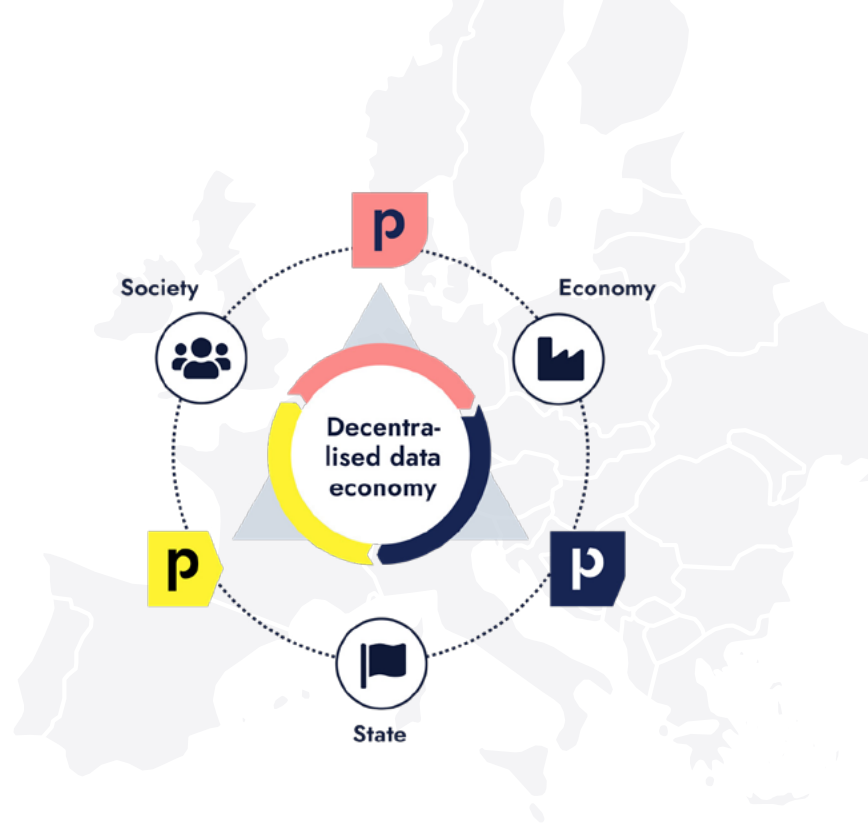
---

... to create something truly revolutionary;  
something that has never been accomplished  
before. Grab a pen. Help us write some history!

→ [polypoly.coop](https://polypoly.coop)

# We're building the European data economy

The polyVerse is the structure supporting polypoly: Three companies working together to build a decentralised data economy – from Europe, for Europe!



## polypoly.coop

**polypoly Cooperative:**  
Represents the interests of all citizens

The core idea of every cooperative is achieving more together. Together we can reclaim sovereignty over our data! We are developing the technical basis for a decentralised data economy: The polyPod, which belongs to all members of the Cooperative. The polyPod is available to all citizens enabling them to reclaim their data sovereignty. The polyPod stores user data on their personal end devices, which it never leaves. Citizens can choose how to make their data available, whether as a donation or for rent. If money is exchanged the cooperative receives a small percentage which is distributed to all cooperative members.

## polypoly.com

**polypoly Enterprise:**  
Creates solutions for entrepreneurs

Successful digital business models are based on data. But Europe is losing its data capital to foreign monopolies. European companies are currently dependent, gradually being deprived of access to business models that drive the digital economy. The solution: A decentralised data economy. polypoly Cooperative provides the technical basis – the polyPod. We develop tools and products for an easy transition to a decentralised data economy. Entrepreneurs will be able to liberate themselves from data monopolies, therefore protecting their existing models and building new digital ones.

## polypoly.org

**polypoly Foundation: Understands the needs of public servants**

Europe is losing its data capital to foreign monopolies. As a result, European companies are cut off from their customers. Citizens data privacy is unprotected, and European states are losing taxpayers' money. We assist governments to build a decentralised data economy for Europe. The technical infrastructure is provided by a cooperative of European citizens using the polyPod. The GDPR is baked into its code. The polyPod efficiently protects European citizens and frees European companies from dependence on data monopolies. It then follows that taxes on profits generated from European data capital are returned to Europe.



**Kontakt:**  
hello@polypoly.coop  
polypoly.coop



**Kontakt:**  
hello@polypoly.com  
polypoly.com



**Kontakt:**  
hello@polypoly.org  
polypoly.org