Portal Master Plan

Max von Tettenborn

October 2, 2023

Contents

Overview

TODO: lay out chapters

Intro

- Each single benefit of Portal can probably be had in some other way
  - It is the combination of benefits that is new
  - Triangle of computer usage
- For some technological improvements, you first have to backtrack
  - this makes the improvement inferior for some time
  - example electric cars

Problems that need solving and how they work

We are currently organizing our digital life in a certain way. We did not consciously choose it that way, instead it evolved over time. Most long-term developments happen like that. However, over time, many problems crept in that we did never properly solve. Instead, we either accept and normalize them (e.g. targeted ads) or we only treat the symptoms by adopting a protective stance (e.g. password managers).

Solving those problems at their root cause would require a more fundamentally different way of organizing digital life. This is what Portal attempts. Answering the common question that start-up founders get asked - “What problem does your product solve?” - is therefore not easy, or at least not quick. It improves the way we organize data and applications and that is deeply baked into the digital life of almost every individual or business.

However, we can provide a list of several of the most prominent problems that Portal solves, or rather, that it prevents from becoming problems in the first place.

Friction in everyday digital life

When energy is lost to friction, it means that a part of the energy you put into a system is spent in an unproductive way and is no longer available for the actual task you want to do. Digital systems have friction, too. It is the time a user must spend on tasks that are needed just to keep the system running properly. With today’s systems, that friction is high and can and should be lowered.

First, think about what your actual tasks are when working with a computer: communicating with others, consuming or creating media, programming, gaming, and so on. Next, think about all the things you are doing to be able to do those things: managing passwords (creating, changing, entering them), transferring data or files between devices or people or applications, doing backups, updating software, upgrading or replacing hardware.

All the time and energy spent on the second category is unproductive friction. We deal with it by using crutches like password managers, file sync services, or we avoid the things that introduce it, like switching between different tools for the same task (this leads to vendor lock-in). Some just have to be done, e.g. backups, updates, upgrades. Of course there is always the option of just not doing it and taking the significant risk of losing all your data or having your identity stolen.

The way Portal works resolves all those points of friction mentioned above. Many can become fully automated, no user intervention required, e.g. backups. But for some problems, it gets even better: they simply no longer apply. Examples are passwords or data sync between devices.

TODO: add links to the solutions

Identity of Humans

Each person is unique and physically exists only once. However, in the digital realm there are lots of copies of us, many entities that all represent a single person. Try and count the number of logins you have, one for each platform or service or social network. Each of your accounts is a different representation of you, each one (hopefully) with a different login.

Maybe your email address is the one connecting factor and you are represented by it? So how many email addresses do you have? And are you maybe logging in to some services with Google or Facebook? Those are more sources for fragmentation. And anyway, email is a specific use-case, conceptually, it makes little sense to also be the thing that uniquely identifies you. This pattern is a crutch that was at hand when needed and got normalized.

So there is no single concept that is your identification in the digital realm. That makes it difficult to prove to a server on the internet that you are really you. The solutions are mainly passwords with all of their problems.

TODO: link to solution

Surveillance Capitalism

A lot has been written about the dangers of surveillance capitalism. In short: when using platforms and services, users are intimately tracked - where they click, what they watch, where they scroll past,

where they pause, and so on. Tech companies learn a lot about every individual (interests, behavior, life situation, …) and use that data to get them hooked on their content. By filtering and presenting content just right, engagement (or rather addiction) can be maximized. This process is totally automatic and self-learning. The goal is to put each advertising message only in front of the eyeballs of those that are most vulnerable to it.

Of course, all advertising is manipulation, some more malicious than others. But the algorithm does not differentiate between harmless and harmful ads. Examples for the latter are political hit-pieces, conspiracy theories, or fake news. And so the case can be made that targeted advertising played a role in some of the most harmful political decisions of the past like Brexit or Trump.

What can be done about it? Look at the start of the value-chain of surveillance capitalism: tracking of users. This is where it all begins. The data that is gathered by deep surveillance is the fuel for all of it. Without it, it would be unknown which ads should be presented for maximum effect. Each user would be a blank page. That is the desirable state we should strive for.

But users can only be tracked because they have no choice but to enter the realm of the tech companies which means using services that those companies host, being a fly in their web. So the first step is giving people a choice about it, allowing them to do all the things they are used to but on their own terms, inside their own home. Once deprived of their fuel, the steady stream of behavioral data, manipulative services will lose much of their power.

Multi-Device Fragmentation

Until the late 2000s, computing felt much different. You had your computer at home and on it was your data, organized with a file-system, and the applications that you installed. This was your primary digital device that tied everything together. But then, the first iPhone was released and soon, smartphones gained more and more significance. For some people, they even replaced the PC as the primary device.

This change in device preferences made everything much more complicated. Since people expected to access data and use apps on all of their devices, they had to be organized in a different way. Apps moved out from peoples owned devices to the cloud so you could access them with your account from your PC and your smartphone. This made surveillance possible as described above. Since those apps could not easily access your filesystem and now you had a separate filesystem on each device anyway, your data moved in with the application itself, also to the cloud. It, too, could now be surveilled and it was now bundled with the application, creating silos and further reducing the control people had.

So the fact that people started using more than one digital device started a process that a) made many things more complicated and b) made people give up control over their digital assets and c) made it easy to scrutinize and manipulate them.

If we could go back to using just a single device, we would be far less dependent on centralized applications we have no control over. There would not be the need to put data out of our hands, just to make it available on different devices. But surely, this is not possible, we need different devices. On the go, we need the lightweight and compact smartphone, when working, we want the powerful PC with large screens.

However, Portal offers a way to have both: multiple physical devices to adapt to different situations and a single platform owned by the user themselves.

Requirements

With Portal, we attempt to create a new fundamental base layer for computer and internet usage. On top of that, applications should be built that cover everything people want to do. It is worthwhile to make a list of those things and evaluate whether or not Portal can in fact replace the current practice.

- Communicate: this covers email, messaging, chat, and audio and video calls. It can be one-to-one, many-to-many or one-to-many. Participants usually need a kind of identity like a phone number, email address, chat handle. Often, the kind of identity is specific to the channel - you cannot call someone on their email address. This implies the opportunity to unify identity.
- Consume: the internet is full of things that can just be consumed without interaction. This is basically the concept of the internet pre-Web 2.0. Think of static websites or streaming services. Although consumption of user-created content also fits if the creators and consumers are not the same demographic, e.g. YouTube, Wikipedia.
- Present oneself: people have profiles on many different platforms like Facebook, LinkedIn, Instagram, etc. so there is clearly a need to present oneself to others, to build an online identity. These profiles should be always online for others to see. They should allow others to initiate contact. And for different contexts, different profiles for the same person should exist, e.g. for business and private purposes.
- Work/Create: Computers have always been tools for creation, whether it is writing, drawing, animating, etc. and a tool to assist with work, e.g. with financial statements, taxes, programming, designing. Some of these use-cases require a powerful machine with lots of expensive hardware or lots of storage space, others do not. Often, work is very collaborative and blends together with the communication requirement.
- Gaming: humans want to have fun and gaming has been a major use-case of computers since their very beginning. Today, a vast range of games exist, some very quick and casual, others deep and with huge time requirements. The demands on the hardware to run them are equally varied. Many games have a social component and so people often have a game-specific avatar and identity.

Technical Description of Portal

To understand the project that is Portal, we need to understand the product that is Portal. What follows is a description of the product itself with some - but not too much - technical detail. It might not immediately become clear how this meets the aforementioned requirements or solves the aforementioned problems. That is normal. Some implications are indirect which is why they will be explained further down.

Portal, the thing itself

Each single Portal is a computer. It behaves like you are used to from computers like a notebook or a smartphone: there is an operating system with a graphical user interface which you can use to manage all things about it. You can use it to install and run applications that were created by others. Each app has its own GUI and its own logic and runs only for you. You can store data in the shape of files on your Portal, as can the apps. So in many ways a Portal is something that has been around for a few decades now.

But: a Portal has no hardware you can touch. That sets it apart from any computer that consumers are used to. Each Portal runs on cloud infrastructure, its hardware is virtual. The implications of this are huge and they are all discussed later. To make it short: Portal instantly beams itself to any physical device you want it to and it has superpowers that come with the cloud like being managed, always-on, or upgradeable with a single click.

It is important not to confuse Portal with a SaaS app. Most SaaS apps also use virtual hardware, but they utilize it as a tool on top of which the developers build the app. That means you share the hardware with all the other users. Only with Portal you exclusively own the virtual hardware. This is like staying at a hotel vs. your own apartment or taking taxis vs. your own car.

Portals have names

Each Portal has a unique identity from the moment it is created. This identity is a random alphanumeric string, the kind of identity computers are good at working with, humans not so much. For humans, we shorten it to the first six digits. It is still random, but easier to remember and to type, similar to a telephone number or an android from StarWars, and it remains sufficiently unique. (Technical detail: a Portal's ID is the hash of its public key.)

An example for a Portal ID is: c0p3x5.

You might wonder why we introduce such a complicated thing to Portal and the decision is surely not done lightly. It is the simplest approach that satisfies several conflicting requirements from the areas

of security, cryptography and ease of use. Further discussion of this decision is a topic for a future publication.

Accessing a Portal

As mentioned above, a Portal beams itself to any physical device its owner (and that of the device) wants it to. Since it is a virtual computer living in the cloud, the owner cannot directly access it and needs to use other devices as middle-men. But those devices are just empty shells. Compare them to a keyboard and monitor that you plug into your desktop PC. Even though you mainly interact with them, they are not the PC.

During the prototype phase, browsers are the empty shells that Portal beams itself into. In other words, every Portal has a web interface you can access. You can find the web interface of the Portal above at https://c0p3x5.p.getportal.org. (Note the ID in the web address.) A browser is well suited for that task because it is specifically made to provide a sandboxed environment where the user interface of a remote service can be displayed. Still, there will be client applications later and they will greatly enhance Portals' capabilities.

Pairing physical devices

What is the act of plugging in periphery for a PC is pairing a device for Portal. The process is very quick and painless and done in seconds. You need to have one device at hand that is already paired. Then you use it to scan a QR code on the new device or vice versa. That's it, the complete Portal is now accessible from the new device.

You can do similar things today with some SaaS applications like messengers that allow you to pair more client devices. But with Portal, this happens on another level: you pair to the platform, not the app. Once paired, all apps on Portal become available, no need to pair each one separately.

Apps

What makes smartphones so versatile is the ability to install apps and that anyone may develop them, so even for niche use-cases there are apps in the store. The smartphone offers a platform for apps to run on and makes all the bells and whistles available to it. This clear separation of concerns - common functions in the platform, specific logic in the app - has been a huge success.

Portal replicates that. However, Portal's architecture adds new advantages to the concept. For example, instead of installing apps on a device (or multiple devices), you install them on Portal which means they instantly appear on all paired devices. Install once, use everywhere. Also, since people only need

a single Portal and because of its cloud-superpowers, there are more common functions that can be part of the platform - and don't have to be part of the app. Examples are contacts/friends, end-to-end encrypted communication, and access control. And then there is hardware: a smartphone app can access the smartphone's hardware (read position, use camera, vibrate, etc.) but a Portal can use all the paired hardware at once. There are lots of new use-cases to be explored based on those capabilities.

There are already hundreds of Apps

Every Portal app is a collection of docker containers, something you would start up using docker-compose. One of the containers must publish a normal web interface. That is all that is needed for a basic Portal app and the important implication is: there are hundreds of such apps out there. The selfhosting community creates and maintains lots of open-source applications that work exactly like that. They are meant for people with technical expertise to host on their own servers but many of them work just as well when used as Portal apps. That means that there is a large offering of suitable apps available right now and the prototype app store is therefore already well-populated.

Peers

One of the weirder consequences of the current way we do things is the coupling of messengers with encryption methods. WhatsApp, Signal, Telegram all come with their own encryption schemes baked in and even promote them. As if those two things - messaging and encryption - have anything to do with each other. They don't, or rather they shouldn't.

In a better world, encryption (and contact/identity management, which is tightly coupled with encryption) should be part of the platform and messaging would be an app that only uses this platform feature. Many other apps would also use the same feature for communicating securely with others. No app developer would have to take on a daunting task like encryption for a relatively trivial use-case like messaging.

Portal does it that way. A Portal owner establishes connections between their Portal and those of others. This is aided by the Portal IDs mentioned above: they are good at preventing spoofing and are ensuring that the end-to-end encrypted channel that can now be established is cryptographically sound. This channel can now be used by any app with a simple API call. What was once only available to app developers with significant resources is now built-in for everyone.

Good old filesystem

In our opinion, a filesystem is an underrated feature of computers. It once was the single place where you can organize your data but newer paradigms pushed it in the background or got rid of it altogether.

Mobile apps hide their file structure away and SaaS offerings don't let you touch it at all. And yet, a filesystem is such a natural way of arranging data e.g. by project or by topic. It also lets you collect files of different types together in one place. With current SaaS apps, this has become completely impossible.

It is also a simple way for apps to interoperate, just by giving multiple apps access to the same files. An image gallery, a social networking app and a photo manipulation app can all share the same image files without the user having to send them back and forth.

For these reasons, Portal embraces the file system and makes it a first-class citizen again. It is one of the cases where making the inherent complexity transparent is actually simpler than trying to hide it behind leaky abstractions. And it is in line with the idea of giving power to the user: the power to organize their data on their terms.

Public and private Views

Since a Portal is always on and always online, it can take the role of the public online presence that is currently played by social networks: an online profile, visible to anyone, with information about a person and a kind of timeline. Equivalents of Facebook, Twitter, Instagram, or LinkedIn could exist as Portal apps. They would publish your information: some of it completely public, some only to a select group of peers. Others would access the content with their browser or through the same app on their Portal. The app could also compile a timeline of all the people you follow, one that is curated only by you.

With the decentralized nature of Portal, this is all happening peer-to-peer and end-to-end-encrypted and without a middleman.

But we can go further than replacing existing ideas. A Portal can provide a public view that is the single place for interacting with the owner. Like a business card on steroids. Someone just needs to know your Portal and can access this page. It would allow them to see your information but also chat with you, call you, book an appointment with you, and so on, all from a single interface. If they access it through their own Portal, they might get privileged access based on their identity, like booking appointments at odd hours.

Noone wants to manage their own server

In the early days of the internet, people actually thought that everyone would manage their own server. However, doing this has remained complicated for too long and internet adoption has spread too fast to non-technical demographics, so a market gap was created: a demand of active participation met a

lack of supply of easy ways of doing so. Centralized services came to the rescue and started providing the means of participation using their platforms.

Unfortunately, we quickly became used to it and what could have been a temporary fix until having your own server became easy enough for everyone remains the only way to participate. So in a way, Portal picks up this unfinished task of making it a no-brainer to have your own server, basically by managing all the technical details for you.

Among the things a Portal owner does not need to think about are: choosing hardware, replacing hardware, upgrading hardware, making backups, managing DNS, managing certificates, securing anything, synchronizing data. Portal does all these things on its own, so you get all the benefits of your own server in a package that is as easy as a smartphone.

Wrap Up of Digital Lifes

To sum up what Portal is: it is a computer that works similar to what people are used to, but lives in the cloud which gives it superpowers. No technical skills are needed to use it, so it achieves all three properties of which other systems only achieve two: It is usable on any device, it is a private and user-owned space, and it is simple to use for everyone.

A Portal is simultaneously its owner's identity on the internet, their digital archive, their public profile, their digital assistant, and much more with the right apps. Those apps are supplied by third-party developers and Portal makes it really easy for them by handling a lot of common concerns itself (hosting, storage, encrypted communication, etc.).

Through all of that, Portal remains grounded. It renounces bells and whistles that provide no real benefit (like recommendation algorithms) or oversimplifications that make everything more complicated in the end (like letting every app manage their data in a silo).

Ultimately, Portal has the potential to replace all other paradigms of computing for consumers because almost all use-cases can be realized on Portal. And it will feel much saner, more controlled and more sovereign.

Most Important Benefits

For the most part, Portal doesn't allow the owner to do anything brand new that was not possible before. But it changes the playing field on which digital life happens.The benefits it offers are more of a fundamental nature as it removes a lot of the complexities and adverse side effects we got used to over time because we had to.

Oneness

What is the center of your digital self? Your email? Your Facebook or Instagram or LinkedIn page? The stuff you keep on your phone or notebook? For many, this is not easy to answer because all of it is important. But there is nothing that holds it all together, no center, nothing to call home. Just fragmented bits on devices or behind accounts.

It will be a huge relief to finally have a single place for everything. We expect this to be the experience where people will ask why it has not been done this way long ago. In hindsight and when you get used to it, it will seem glaringly obvious.

All of your data will exist only once, your contact list is managed once, every app is installed just once and all of your devices show you one singular view. For all intents and purposes, your Portal will be the only computer you will use, the one that unifies everything.

The overhead that this eliminates is huge and probably amounts to many lost weeks over a lifetime. No more data needs to be copied or synced between devices, no more exporting and importing stuff between web services, we can even get rid of the bane of the internet: passwords! When looking back at today, these things will look like manually cranking the engine to start a car.

And the same is true for app developers. By its nature, it will be a breeze to develop for Portal because so many functions are just part of the platform itself: hosting, data storage, secure authentication, contacts, end-to-end encrypted communication, just to name a few. A developer will use an API to access these functions and not need to worry about them themselves.

Portal really strips away a lot of the accidental complexity for users and devs and lets us all concentrate on the intentional complexity.

Sovereignty

When the internet was adopted by the masses it was open and decentralized and no one really controlled it. We all had high hopes that it would improve relationships between all of humanity. That did not happen and instead it has become a surveillance and manipulation engine much more powerful and dangerous than anything before.

To explain the role Portal can play in this context, we should very briefly look at a bit of history.

It seems like the problem was one of timing. At the moment where the first web (Web 1.0) was established and people were ready to participate online (which would be known as Web 2.0) there was no simple way for them to do that. Some assumed that people would start building their own servers and websites but the expertise and patience required for this made it unviable for most. So there was an unfulfilled demand for a while and it was filled by the first centralized social networks, first among them Facebook. They allow participation and relieve you of the burden of managing the technicalities.

In a search for monetization strategies, they found advertising to be elegant. The service could remain free for users and still make money. It might have been a good idea at first but as we see today, it soon got out of control and today's users are primarily ressources from whom attention is to be extracted as efficiently as possible and whose thoughts and attitudes are for sale.

Let's look back at the fateful moment when people were ready to participate and their only real option was centralization. What would have happened if instead a company made hosting your own server radically simple and that became the backbone of Web 2.0? Would we still have a decentralized internet today? Would people's private data have remained on their private devices? If someone proposed a model like Facebook, would we laugh them out the door?

Perhaps the technology was not ready at the time, but now it surely is. And Portal is the attempt to right that historical wrong turn. If successful, it will be able to restore the balance of power between tech companies and the people.

For many individuals, this might seem like an abstract benefit, not easily grasped. But for society as a whole, it is a crucial improvement. Manipulation of public discourse is immensely dangerous to our ability to find solutions to our most pressing problems. There are credible reports that link targeted manipulation on Facebook to the Brexit vote and Trump's election, two incidents that we consider extremely harmful. And what fuels the manipulation is surveillance. Only if you know an individual's behavior intimately, can you decide which ad to show at what time in what context for maximum impact.

Wide adoption of Portal turns off that flow of private data and starves the machinery that relies on it.

Beyond physical devices

We are many years into the era of cloud computing. One would expect that physical devices are unimportant and easily interchangeable by now. But surprisingly, they are still extremely important and many people store data exclusively on their PC or always need to find their smartphone when using a specific app.

The promise and expectation of cloud computing paints a picture of data and applications that are just somehow there, almost independent of any physicality. You could just pick up any device and let it be your gateway to all of your digital life. This has not materialized for several reasons.

Firstly, there is no all-encompassing piece of cloud for you to use so each app has its own little cloud-backend where it is separated from other apps and behind a separate account. What could have been a collection of apps in one space is a collection of services with all the additional friction that comes with it. When accessing something from a new device, instead of logging in to one space, you need to log in to each app separately. Avoiding that effort leads to a kind of device lock-in.

Secondly, people don't trust the cloud and for good reason. With tens of services each of us uses, how are you supposed to tell secure ones from those that cannot protect your data or even outright sell it? Are you supposed to do research on each single one? Better to not trust any one and keep personal data locally.

Portal attempts to solve both problems. It is exactly this all-encompassing piece of cloud that has been missing and that you can use for anything from anywhere with a single login. And it attempts to solve the trust problem using multiple measures like separated infrastructure for each customer or aligned incentives through a subscription-based model.

The original promise of cloud computing can finally become reality. Your cloud - your Portal - can completely replace anything that is tied to the physical world. Your devices are just an empty shell for your Portal to inhabit. And any device can play that role, even if just for a few minutes. This can completely change our habits regarding digital technology, and we will explore the larger implications later.

Implications

Adoption of Portal - in particular mass adoption - offers many more beneficial implications. When thinking it through, it is surprising how many things work better just by arranging the digital primitives more naturally.

Implications for Portal owners

If you own a Portal, your digital life is pretty different from what it is today. Mostly because so much friction is just gone and things just work as you expect them to, as it is obvious. The individual benefits are what motivates people to own a Portal.

Identity is easy

Earlier, we described that there is no real identity to your digital self. The thing that comes closest is your email address and that is still awkward since writing emails is just one simple use-case that now somehow has to fill the shoes of symbolizing your identity.

But your Portal can fill that role. It is yours, personally, so there is a one-to-one relationship between your Portal and you as a person. It has an identity that is fit for computers to authenticate (its public/private key) and relatively easy for humans to handle (in the shape of the six-digit Portal-ID). So the Portal can act on behalf of its owner when contacting other Portals or accessing resources on the internet and always prove its authenticity. This gets rid of passwords!

If you think about it: the necessity of passwords is another byproduct of having multiple devices. If we would use just a single device all the time, it could just save the password upon registration - similar to a cookie - and we would never have to even see it. Only the need to prove our identity from other devices creates the need for exposing users to passwords directly. Portal is exactly this single device besides which we need no other. That is why it can represent us so well.

TODO: describe grouping of peers and badges that can be given away. Both can be used for access control.

2FA is easy

Of course, passwords are also weak. Over time, many tricks have been developed that allow attackers to learn passwords. For that reason, the inconvenient crutch that are passwords was extended by another even more inconvenient crutch: 2FA codes. They are regenerated every few seconds and you need a device or app that handles that and which tells you the currently valid code for your login.

And of course more and more web-services adopt the practice and so your authenticator app is filled up with tens of 2FA codes, just like it happened with your password manager before. We just doubled the cognitive load of authentication.

It is worth noting that machine-to-machine authentication is pretty easy and extremely secure if you use public/private key pairs, like Portal does. As explained above, Portal can authenticate itself with any other system it is in contact with. So what remains is the Portal's owner authenticating themselves with their Portal.

Here, we can use all kinds of methods: device pairing, unprivileged pairing for untrusted devices, two-device confirmations for critical actions, recovery codes that are printed on paper, and so on. The thing is: a user has to do this all only with one service, their Portal, not with tens of services like today. And that shrinks the cognitive load significantly.

Syncing between devices

This problem is just gone. Not solved - gone. It does not appear at all if you use a Portal.

There are no different devices since your physical devices are just empty shells that connect to your single Portal. It is the single source of truth. Whatever apps you installed there, data you saved there, it just appears on all of your devices simultaneously.

Device-specific apps are gone

A corollary of the omni-device nature of Portal that is still worth mentioning separately is the reduction of device-specific apps. Those are the apps that exist only for one platform: PC or mobile. Whenever you sit at your desk and still have to get your phone to do something you are probably using such an app.

As soon as someone fully switches to Portal, these will be gone. Every app on Portal is cross-platform by nature, making it easier for users but also for developers who don't have to maintain multiple versions of their app.

There will probably be exceptions, e.g. apps that make no sense without big screens and mouse/keyboard input like 3D modeling applications or certain games. But since the vast majority of today's web-apps works well on many device classes they will, too, on Portal.

Fully automated backups

It is a bad idea to have no backup strategy for your devices. But implementing a good one needs some work and there are some details one can easily get wrong. You will want an incremental backup, so it runs fast. Some older backups should be kept in case you need to retrieve something you deleted a while ago. But they should be deduplicated or else the needed storage will be huge.

Setting this up and managing it is not productive work, it is friction. But it should be done. The heterogeneity of platforms and devices prevents a good one-size-fits-all solution.

But of course, Portal can do it all for you, out of the box, you don't even need to activate it, you don't even know it is happening until you want to retrieve it. We don't call it a fully-managed service for nothing.

Portal's backups have all the properties described above and they are encrypted client-side which means the storage that contains them (which is of course not the Portal itself) only ever sees encrypted data. If needed, you can download a full backup of your Portal for a given time. We will keep these even if you cancel the service, so your data will not get lost.

Contact management

It is normal for every operating system that some concerns are solved by each app and other concerns are solved by the OS so all apps can benefit. The latter are usually low-level things like managing network traffic or drawing pixels on a screen. The app tells the OS what to do using an API.

When Portal is used at the owner's main platform, the set of concerns that Portal can manage grows. Now it also contains things that usually get out-of-sync when using multiple platforms or web services.

One great example is contact management: the list of people you know and whose contact details you keep.

Think for a moment how many such lists you are managing right now. You probably have a phonebook on your phone and that might be the main one. But then, you may have friends on Facebook, LinkedIn, Twitter, and all the other social networks. Are all these lists in sync? Probably not.

Portal controlling devices

We have covered in detail how a Portal can be controlled from all of an owner's devices but the reverse is true as well. When pairing a device with your Portal, the Portal can now control the device. So each paired device extends the range of capabilities of the Portal. Here are a few examples how this can be useful:

- The Portal can determine which device the owner uses or is close to at any moment and route notifications to that device only instead of triggering a crescendo of notification sounds everywhere.
- Apps can use multiple devices at once, e.g. display media or information on one device while allowing user interaction with another. This would be similar to the ChromeCast feature but with fewer restrictions. You could use a fixed device in a meeting room to display and control a slide deck from your Portal. Also games could make use of this pattern.
- Running interactive sessions, e.g. a phone call, can be smoothly transferred between devices. When coming home, you could transfer an ongoing call from your phone to a set of speakers or your TV.
- There surely are many more ideas that app developers will think of.

Implications for society

From a bird-eye view on a whole society, Portal brings many benefits as well. These are the effects that are in fact most important to us since they have the potential to facilitate real, much-needed change in many critical areas. But history has shown that the greater good is not a sufficient motivation for many people to buy and use a product. That is why benefits for individuals are needed. They pull people in. The more important benefits for society are achieved almost as a side-effect.

Identity is easy (part 2)

Famously, "on the internet, nobody knows you're a dog." What was once a fun point of a cartoon has evolved into a real problem. But dogs are not the problem, bots are. And as fake content gets more

sophisticated and is mass-producing convincing audio, images, and video, we face an epistemological crisis we are nowhere near prepared for. How is the average person supposed to tell apart authentic from inauthentic content?

Portal might just be a puzzle piece for a solution. Proving the authenticity of a piece of data is mostly a solved problem in cryptography at least since the 80s when asymmetric cryptography was developed and is an integral part of any modern encryption scheme. By digitally signing data, one can prove that it is authentic in the sense that the signer approves of it in some way.

The problem is that the signer is a pretty abstract concept, because it is in essence a cryptographic key. The key is associated with some real-life entity like a person but now proving the association between key and person is the real challenge, particularly from a UX perspective. At one point, people imagined regularly coming together to key-signing parties in order to check and verify these associations. This of course never has remotely reached the mainstream.

Portal however has your cryptographic identity built-in in the shape of your Portal's ID. Simple as a telephone number, prominently visible in the UI and cryptographically secure, this just might prompt people to get used to identifying others in that way.

That in turn would allow people to prove their authenticity and that of the data they share to others in a verifiable way. It would give people a stamp of approval they could attach to data, allowing them to mark authentic material and allow others to tell it apart.

More economical physical devices

When fully embracing Portal, your physical devices are just an empty shell, as said before. One implication of this is that the device need not be as powerful as they are now. Most of the heavy lifting regarding processing and data storage is done by the Portal, the device only renders the UI. Devices can be more economical, cheaper, less resource-intensive and also live longer before they need to be replaced.

Also, sharing of devices is becoming more viable because pairing and unpairing is so quick and easy. If you only need a tablet or a desktop computer from time to time, it might be ok to only have one in a family that everyone may use. Floating desks in companies that embrace a hybrid office model will also be much easier to implement.

Ultimately, this might lead to a more conservative use of physical devices and the resources and processes needed during production.

Decentralized personal data

No system is perfectly secure. There is always a risk for bad actors to find and exploit a vulnerability. We have seen many services over the years that have been hacked and where user data was stolen. There is no reason to assume Portal is different.

However, Portal has an advantage: personal data is not stored centrally in a single storage system. It is distributed among all the individual Portals, each a separate piece of infrastructure. If one Portal is compromised, only the data of one individual can be stolen. There are natural moats between all users. We regard this as an important defensive position that reduces the scope of any data theft.

Distributed infrastructure means distributed power

There is no denying that putting the means of mass conversation in the hands of private companies has been a mistake. We describe the fallout above. Unfortunately, there are not a lot of alternatives, at least none that have caught on.

After all, any alternative must provide hosting for persistent user profiles, messages, media, and so on. And until now, the default method to achieve that was putting it all on more or less centralized servers because no one wants to manage their own server. But that also means that the power over the conversation is centralized.

Portal offers the infrastructure that is needed for mass conversation but it does that in a truly decentralized way. Now, it is radically easy to manage your own server. There is no place for a gatekeeper, no one can subtly nudge public conversation.

It remains to be seen if conversations become better with that model. They will surely become more authentic and people will be less prone to be thrown into extremist rabbit holes in order to maximize their engagement. We hope that this goes some way to mitigate the ideological rifts that have been growing so rapidly in recent years.

For developers

Portal is a two-sided product. On the one side, there are owners of Portal, our customers. On the other side are developers that adapt or create applications to run on Portal. Both sides complement each other. So it is important to make developers' lifes as easy as possible and Portal is well-equipped to do that. Like with the owners' experience, the main benefits lie in the things that need not be thought about at all.

Hosting and Infrastructure

Today, if you develop any app that is not totally self-contained on a single device (only very few are), there needs to be some kind of backend. This should generally be always available. So the task of an app developer is not only to build the app, but also to continuously provide the backend and manage any maintenance tasks that come up. If they fail or for some reason stop to do so (e.g. lack of interest or bankruptcy), their app might lose critical functionality or stop working completely.

Not so on Portal. Each individual Portal provides the infrastructure an app can run on. As a developer, you just have to publish the app. There are no additional ongoing obligations. Features that currently are only possible with a continuously running backend - like multi-device sync or a permanent online presence - can now be achieved by a built-once application.

All things crypto

Security and cryptography are complicated topics. But for many use-cases, they are unavoidable. Whenever you manage public-facing infrastructure, you need to concern yourself with certificates. And if you want to give users the option to talk to each other with a high degree of privacy, you need to build end-to-end encryption into your app - a very difficult undertaking!

A side-note: the fact that chat apps need to be bundled together with their own encryption is an affront to all good software engineering. Chat and encryption are two completely separate problems, they should be solved separately. The reason this does not work is that encryption relies on identity and there is no persistent cryptographic identity for people, so each app has to have their own.

With Portal, those difficult problems are completely managed by each Portal itself. The developer need not even know about it. Portal offers an API that allows them to send encrypted messages to other Portal and authentication is handled by the Portal's own certificate. It even checks all incoming connections to an app and makes sure they are allowed and if they originate from a paired device, another Portal or are anonymous and labels them as such. Each app can then simply decide what to do with them and focus fully on the actual business logic.

User Management

When building a web-application or one with a backend, you need to manage your users. Each one needs an identifier (probably an email address) and a password at least. Perhaps, you mandate 2FA. You need to make sure that every request is properly authenticated and authorized.

Each instance of a Portal app only serves a single user, the one on whose Portal it is installed. For many apps, this means that a whole lot of complexity just vanished. And if an app needs communication

between users, Portal offers a simple API that allows it to query its peer Portals and talk to the app instance that is installed there. Again, all details are taken care of by the Portal itself.

Monetization

Monetizing application development for small companies or individual devs is often not very straightforward. Portal offers a revenue share model which has significant advantages. Most importantly for developers, they have to do almost nothing to profit from people using their apps, apart from telling us where to wire the money. And most importantly for users, they never have to pay anything on top of the Portal subscription fee they already pay. That way, they are incentivized to install any app they like and as many as they like without a second thought.

We describe this model and its implications in more detail later.

Business Model

Advertising has evolved into the business model of the internet. When people expect their web services to be free and setting up a payment process is seen as a major hurdle, there is not much else you can do as a company to cover your costs. And it turns out that once you intimately know your users, you are in a unique position to influence them.

But in recent years, the damage this does became more obvious and well-researched. And people are more willing to actually pay for a good service. There are now many web applications operating on a subscription model. This might be due to privacy concerns. Surely, another part of the reason are services like PayPal, Google Pay, or Apple Pay that offer a smooth experience around the actual act of paying.

This is good news for Portal in multiple ways. Firstly, Portal also is a subscription service so customers' willingness to pay in general means that paying for Portal is at least not an outlandish idea. Secondly, a single Portal is in a position to replace multiple web applications at once because of its ability to install the right app for any use-case. This could actually be a money-saver for some owners who currently pay for each application separately.

Subscription price vs infrastructure cost

Portal users pay for their Portal on a monthly basis, around €15 for the smallest version right now, but that might change as we learn more about the willingness of customers to pay for the service. More performance and storage cost extra and a yearly payment gives a discount. This is all standard practice.

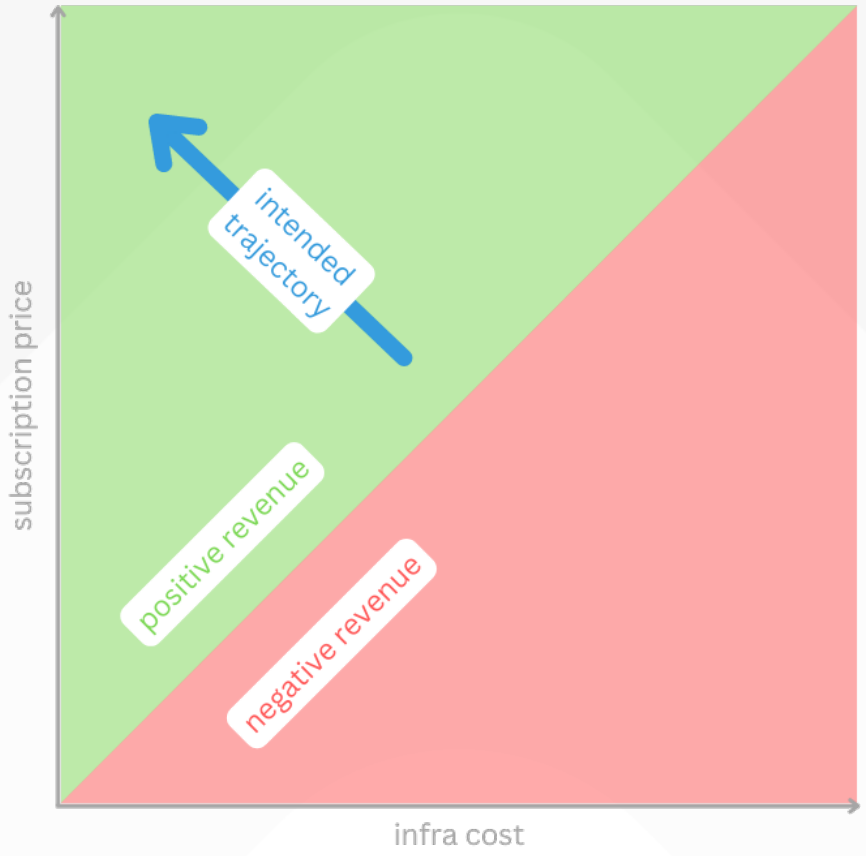

Figure 1: Subscription price vs infrastructure cost

Over time, Portal will become more useful and versatile as more apps populate the app store and more core features are added. It might be feasible to increase the price proportionally. That might happen naturally as people use their Portal for more things and buy more storage and larger-sized Portals. This accounts for the upward trajectory in the graphic above.

We use the monthly payment to buy the infrastructure on which each Portal is running. There is a range of options available to us which will be becoming practical in stages over time.

1. IaaS providers like MS Azure let us provision the VMs backing each Portal right as we need them, priced individually and only for as long as we need them. There is practically zero overhead or up-front investment involved, so we can pay as we scale.
2. Reserved instances on IaaS providers let us commit to VM usage for one or multiple years in advance. This drastically reduces the price but we have to take the risk of paying for unused reservations if demand for Portals goes down. This is best combined with customers who also commit to longer-term plans.
3. Renting bare metal from IaaS providers and managing the VMs on top for ourselves is cheaper than renting the VMs but only if we can sufficiently utilize it and it introduces the management overhead.
4. Colocating one or more complete racks of hardware in a datacenter is cheaper than renting bare metal but now, we also need to take care of the hardware.
5. Building our own datacenter is the cheapest option per Portal but we probably need hundreds of thousands of subscribers to utilize it and it is a huge undertaking.

From stage to stage, the cost per Portal goes down but initial investment and risk go up. This accounts for the leftward trajectory in the graphic above.

Taken together, the revenue per Portal is bound to increase over time.

Revenue share with app devs

Monetizing applications on the web has never been very straightforward for lone developers or small to medium development teams. There are several options and they all have significant downsides.

- Including ads allows the app to be free of additional charges but users usually pay with their data and attention and we discuss above what that can lead to.
- Letting users sign up for monthly plans works only for applications that are really valuable to the user and need to be used continuously instead of once or seasonally. Also, implementing management and payment logic is some effort.
- Selling apps for a one-time payment only yields recurring revenue with constant growth.

Portal offers another kind of revenue model to app developers: revenue share. In short, it works like this.

- Part of Portal owners' monthly subscription payment is a mandatory app flatrate. They need not even be aware of it.
- Each Portal collects installed and used apps for each month and reports them to the backend (not before allowing the owner to censor the list, for privacy reasons) where it is anonymized and stored.
- The app flatrate of each Portal is split among the apps that the Portal reports.
- For each app, the splits from all Portals are summed up and that amount is paid to the app developer.

This mechanism is fully automatic and provides many benefits for all involved parties.

- Portal owners need not worry at all about app costs.
- Portal owners can freely install any app they like since each additional app is always free - it does not incur additional cost.
- App developers need not worry about payment processes.

With revenue share in place, we expect a virtuous cycle to occur, where the number and quality of apps and the number of Portal users drive each other up, as displayed below.

Roadmap

- Endgame is simple
  - Portal should cover all computing needs
  - some stories later
- first steps must be more conservative
  - solely browser-based interaction to save dev effort
- 3rd party apps that were originally built for selfhosting
- important goal: get a self-enforcing feedback loop going
  - users pay for Portal
  - revenue gets shared with app developers
  - app developers see opportunity and improve/adapt/fork apps for Portal or make exclusive apps
  - Portal becomes more useful
  - more users pay for Portal
  - repeat…

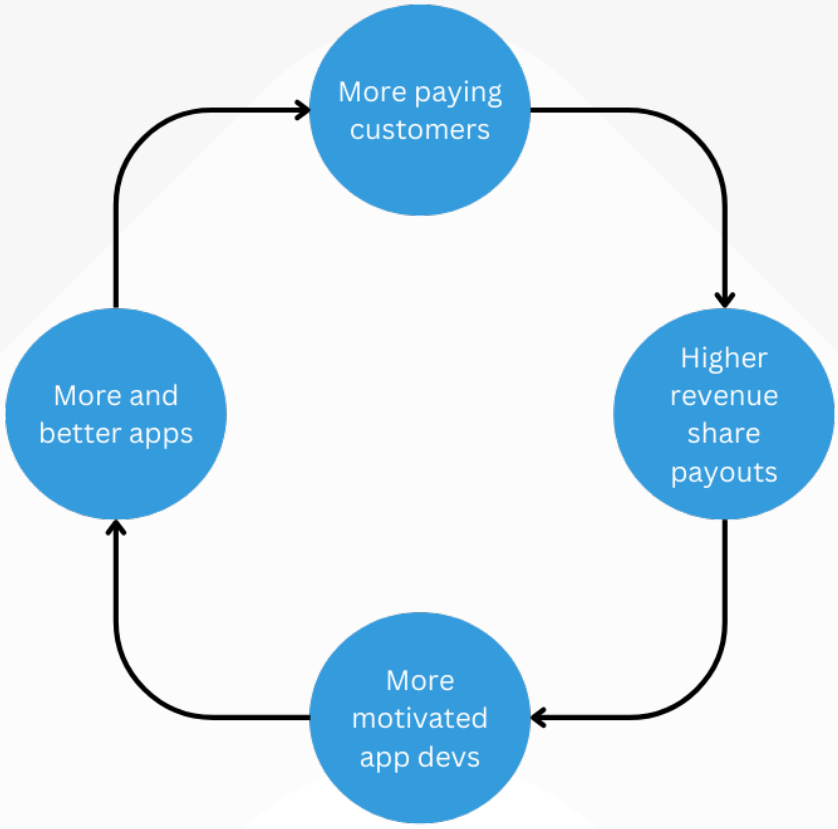

Figure 2: Virtuous cycle of app development

Endgame Uses and Stories

- medical records on Portal
  - A Portal keeps a complete history of medical records
  - records are tagged by category
  - owners can give read-access to specific doctors
    - for a certain time
    - for a certain category
  - doctors can add new records
  - doctors request access so the owner just has to confirm
- public space
  - portal owners can shape the public space of their portal, the one that everyone can see
  - they can put text and media there, make it a profile or a business card
  - apps can offer widgets that can be placed on a public space, e.g. for calling the owner or for making an appointment
  - badged/achievements can be displayed there
    - they are a blob signed by some other Portal, so anyone can make achievements and give them away to others
    - there are official achievements by Portal that are marked with some special symbol
- apps interact with each other
  - no privacy problems since it is all on the Portal
  - novel ways of apps interacting
    - health tracking setting calendar events
    - shopping app knowing about finances
    - meal planner and medical records
- Rental devices
  - People pair their Portal only for a short time with other devices
  - They are paired with limited access and critical actions need to be confirmed from a trusted device
- Payment

Far Future Possibilities

- Micro-Terminals

- Things like Jewelery or Pens can act as your physical device
- They can have a voice interface
- More important: they allow you to take command of other devices in your vincinity which are more comfortable to interact with