We increasingly rely on digital spaces.

> "We shape our buildings, thereafter they shape us", the digital spaces are the buildings that shape our minds. They impact our information and social relationships directly, weave into our collective consciousness and become part of us.

Without a common language, we cannot imagine, articulate, and realize a shared future.

> To retain our agency, digital spaces should be processes we actively participate, instead of products we passively consume. They can be open ecosystems of protocols, software, and people that better ensure autonomy. Yet such potential is hidden in jargon and only understood by a few, and the possible future is beyond the experience for most of us.

We can summurize patterns that aid human flourishing, and form languages to imagine collectively.

> The idea "pattern language" was first used by architects to enable ordinary people to better design and build their physical spaces. The most habitable environment are incrementally built by the inhabitants, but the specialization in design and construction has limited personal agency in modifying our environment. Pattern language is a way to penetrate that specialization, and it might work for digital spaces as well.

By returning agency to non-specialists, more can explore the balance between individual agency and community autonomy.

> The concept of pattern language has already influenced product design and software engineer heavily, especially in the form of design patterns. In applying pattern language to digital spaces, we can hold an intention closer to that of Christopher Alexander et al., to enable the agency of individual and self-determination of communities. Using philosopher Ivan Illich's term, they are "convivial tools" that allow "individual freedom realized in personal interdependence".

Each pattern functions on a particular depth in an ecosystem.

> The effectiveness of a pattern largely relies on its context, its relationship with patterns that limit it and the patterns it enables. In A Pattern Language, this relationship is presented by the spacial scale, and patterns are organized on the spacial scale to describe a particular environment.

It is the depth of an user's journey into the ecosystem.

> For example, a newcomer can start by reading the public content in a digital space, before creating content in the network. Then she can continue to organize a community and host her own server for the community. If she has the necessary skills, she can contribute to the client she uses, or fork one to make her own. Then finally, she can participate in improving the protocols shared by multiple parties, which takes more discussion and a slower consensus-making process.

It is also the depth of time, and the depth of "protocol hardness".

> This is a scale of increasing agency and impact: the deeper you are in an ecosystem, the more influence you have on it. It is also a scale of decreasing rates of change: the shallower layers, such as the topic of content, change fast, while the deeper layers, such as protocols, change slow. This creates shearing layers as elaborated in How Buildings Learn by Stewart Brand, in which the dynamic of the faster layer is determined by the slower, while the slower layers need to gradually incorporate trends in the faster.

This a personal imagination of how such pattern language could look like.

> It follows your journey into an ecosystem of digital spaces, describing 4 patterns you encounter at different stage. You start from ...

Town square to rabbit holes

You walk anonymously in a publicly town square, discovering content and people emerged from different communities. You follow them into their communities, each with its own gating and moderation policy, as well as focuses and social norms.
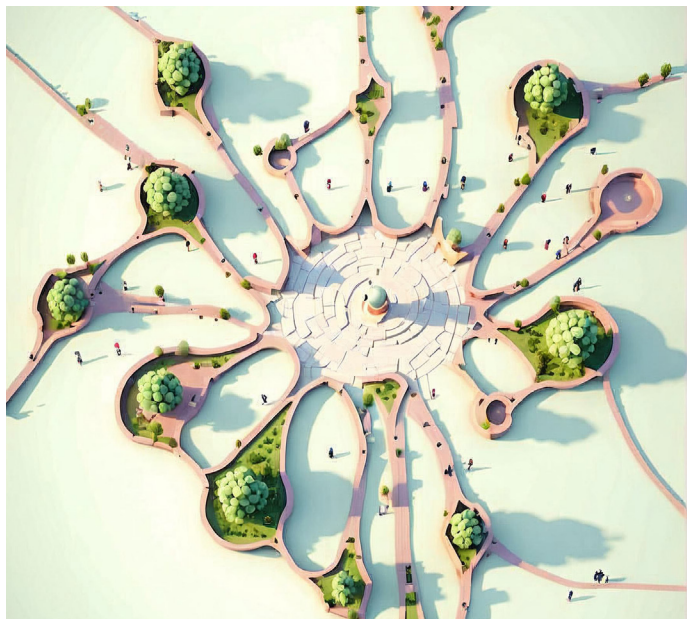
The town square shows the same result for different people instead of providing personalized recommendations. This allow us to freely explore what others are discussing outside our echo chamber. The communities that the town square leads to gather like-minded people and sustain discussions around certain topics, however niche they are.

We can find this pattern on the how people use Reddit to enhance Google search, Reddit and Discourse popular posts rank, 豆瓣话题广场, 微博热搜, and many other products and ecosystems.

Identities based on communities

By creating content and building friendship, you create a new identity, ingrained in the communties you joined and organized. You publish to a community space, following it's moderation policies, and rely on your friends to regain your identity when you lose it.


In starting with a new application, we choose a small group of friends to seed a new social network, which allows us to persevere certain social connections while exploring a new identity.

It has been [argued](#) and [experimented](#) that when we can post only as part of a community, we act more civilized and alleviate the need of content moderation.

In the Fediverse, instances can "defederate" each other, allowing each community to retain a high level of agency in content moderation.

Different application, same content

You explore these communties and sustain relationships with people you meet. You can switch between different applications, but the content remains the same. Applications can hide or highlight different content, but cannot forge authorships.
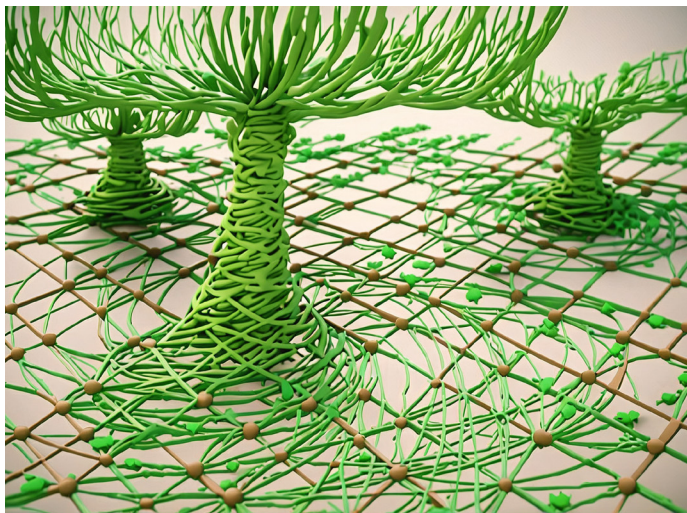
We can switch applications easily when the content we create belong to us, instead of the applications. Applications are interoperable through shared protocols, lowering barrier of entry and creates more competition in optimizing user experiences.

Many ecosystems of social protocols are examples of this pattern, e.g. Secure Scuttlebutt, ActivityPub, AT Protocol, Nostr, Farcaster. In reality, however, the variety of applications and the easiness of migrating between them varies.

We expect to own the content we create even on a centralized platform. So when Reddit effectively banning 3rd party applications by charging for its API usage and, Reddit users still organized an uneffective but massive blackout.

Strong structure, unfinished surface

As you build an application or service, you find that the core requirement of interoperating with others are straightforward. Less common features and format-specific conventions are messy, with de facto standards that you can choose to follow or violate.

Buildings that are well loved and adaptive to generations of changes often have strong structure that endures time, but also unfinished surface that invite hacking and modification. "(H)umans have a taste for things that not only show that they have been through a process of evolution, but which also show they are still a part of one", noted Brian Eno in *How Buildings Learn*, "They are not dead yet."

Alive protocols are also like this. Nostr began with a minimal set of rules, ignoring many pratical concerns. Standarization often come after practices become common place, and extensions such as Lightning Zaps do not affect core interoperability. Other protocols with more comprehensive design started at similar time but did not gain similar tractions among developers.