DRAFT v1.27 redacted

Samuel Stroschein
CEO Opral US Inc.
New York

TL;DR

- We are building a change control system and SDK called “lix”
    - An easier way to build collaborative apps
    - Apps will be more capable (CI/CD, auditing, and change requests)
    - Enables AI due to the ease of data access and auditing
    - Puts cloud-based apps and SaaS like Notion, Rippling, Figma, or QuickBooks at risk for disruption because buyers will prefer lix apps. Lix apps are more capable (change control), have no lock-in (data ownership), and don’t need integrations (interop).
- The go-to-market product “inlang” (an ecosystem to globalize software) is adopted and loved by developers
- Growth can be accelerated by fulfilling user requests faster and investing in marketing

Fundraise

[redacted]

Target investor

Is not monetization focused. The opportunity is building and growing the infrastructure as fast as possible, and not getting slowed down by short-term (next 1-2 years) monetization.

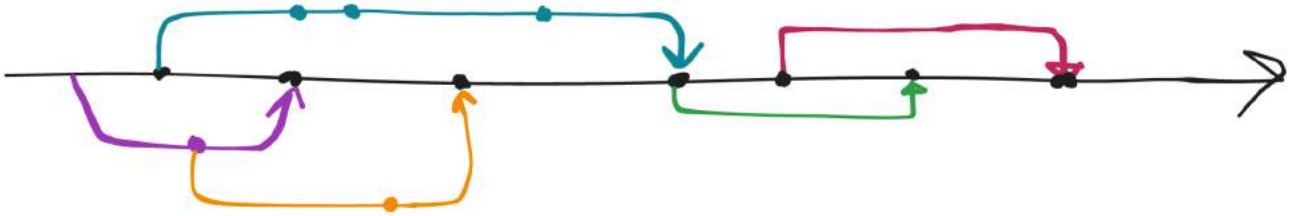
Team

Multiple ex-founders/CTOs + design engineers with mini startups

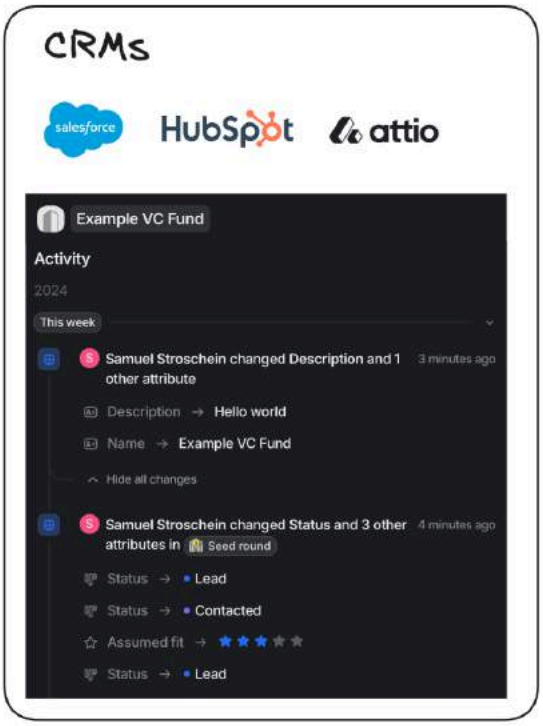
Demo

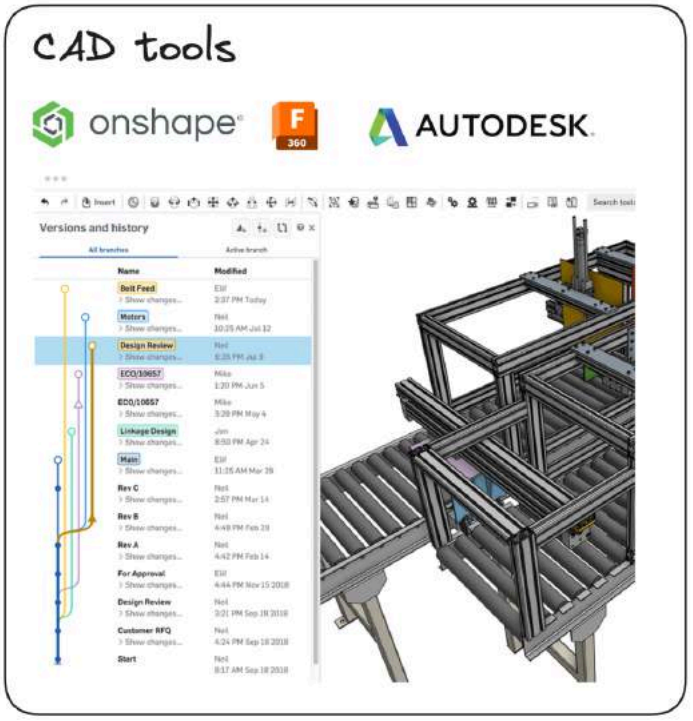
https://www.youtube.com/watch?v=b5EWVgQP3R0

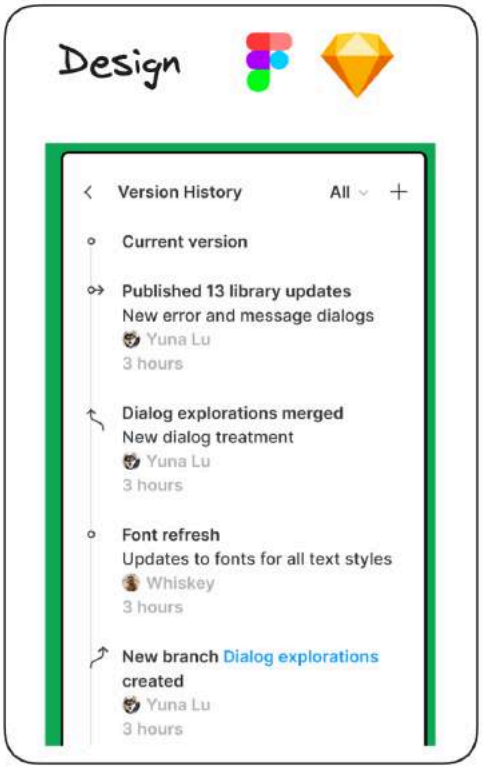
Preface: It's all about changes



Every app and service reinvents the wheel for change-related features such as collaboration, change history, automation, or auditing. The result is lock-in, lack of interop, difficult cross-team collaboration, and a harder time building automation pipelines (CI/CD).







The lix change control system and SDK

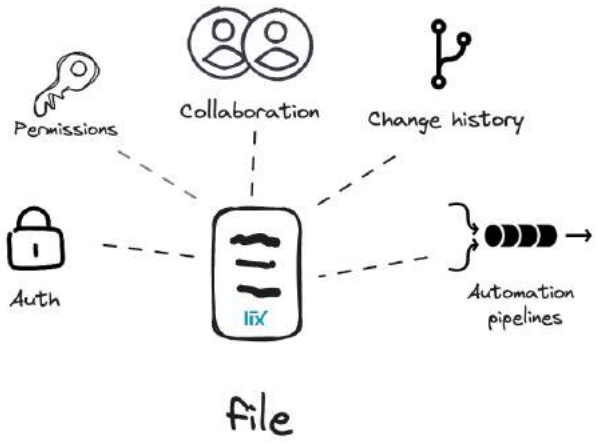
“Git as a backend for apps.”

A change control system is like a version control system (git) but understands changes in different file formats and exposes an SDK to build apps. The SDK allows apps to query changes, trigger automations, and more.

Lix brings backend features to files

A change control system knows who the actors are (auth), allows actors to conduct changes (permissions), be aware of how data changes over time (change history), merges changes (collaboration), and the knowledge of changes can be used to trigger automations (automation pipelines).

In other words, solving change control automatically brings backend features to files. Apps and services don’t need to build custom auth, collaboration, or automation layers.



Lix brings modern backend features to files.

Lix is an easier way to build apps and services

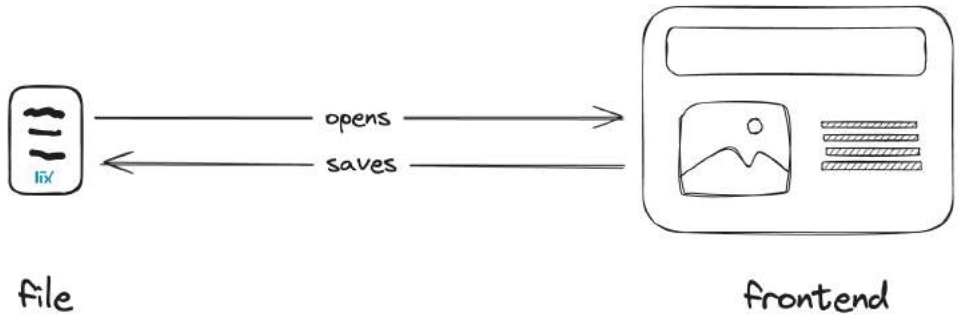
Lix leads to a cheaper and more powerful way to build collaborative apps than cloud-based apps:

1. Define a file format.
2. Write a lix file format plugin.
3. Develop the frontend. Done.

Developers using lix as a backend do not need to concern themselves with cloud infrastructure. Lix enables collaboration, notifications, auth, and soon-to-be table-stakes change control features out of the box.

In contrast, cloud-based apps are expensive to build due to undifferentiated engineering efforts:

- Auth
- Integrations
- Ensure customer data integrity
- Scaling complexity of storing all customer data [Figma, GitLab]
- Replicating change control features like history, branching, etc. [Hex, Lokalise, Figma]

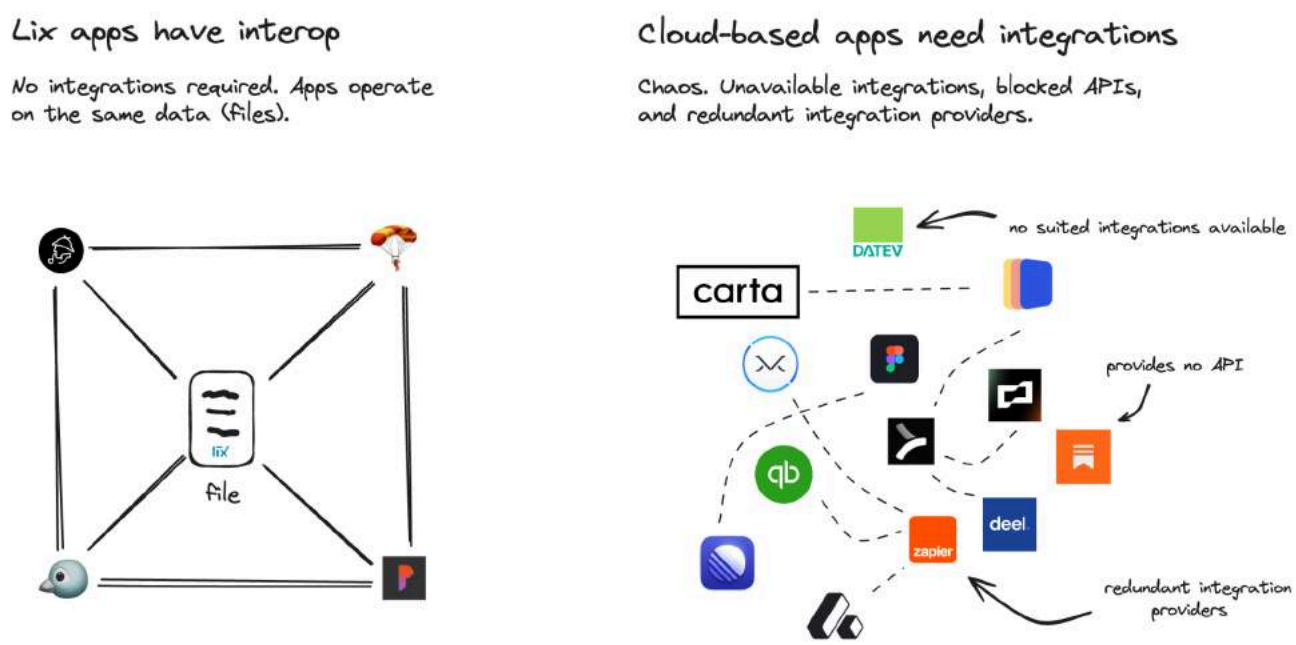


Developing lix apps feels as easy as in 1995: Open a file, develop the frontend, and done.

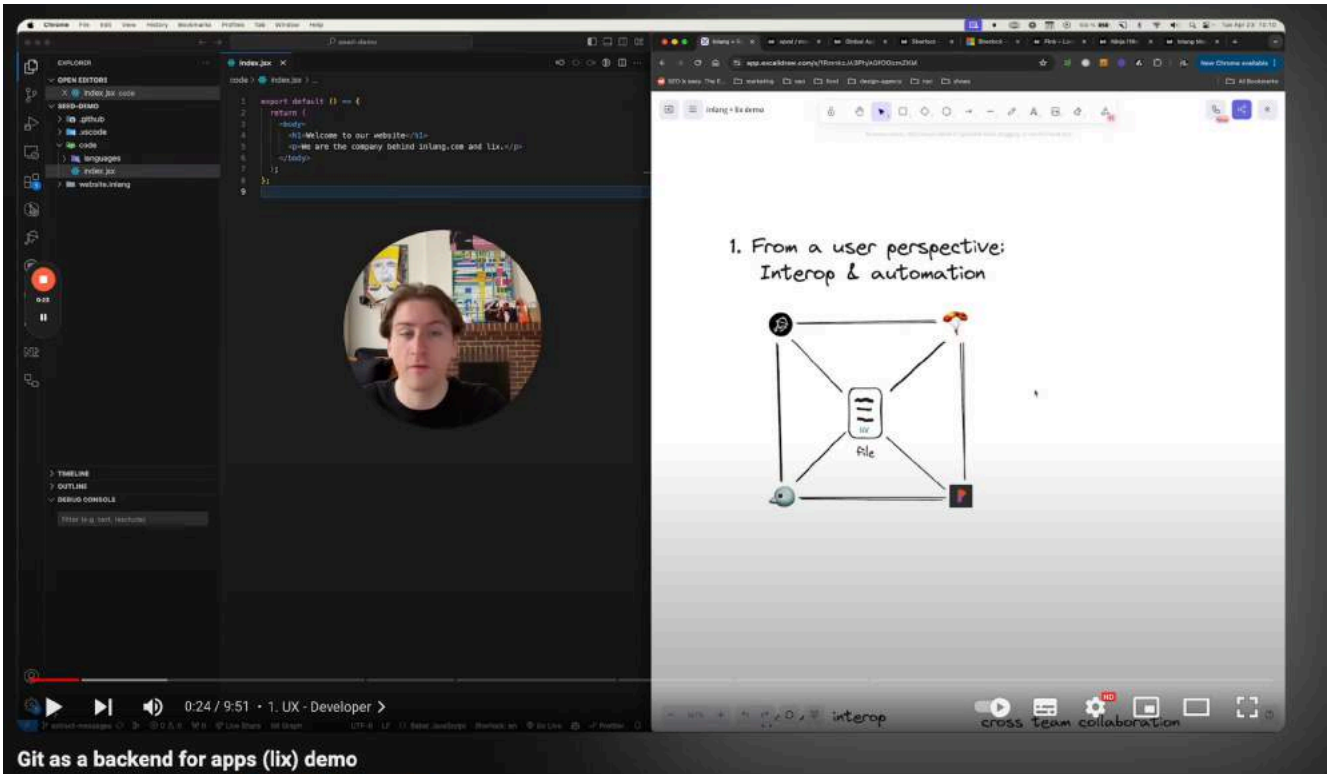
Lix enables more capable apps and services

Interop

Lix apps and services that leverage lix have interop between each other. Any app can open a file and process the data. The need for integrations that transfer data from cloud app A to cloud app B is redundant.



Lix apps operate on the same data, making integrations redundant. Cloud-based apps do not operate on the same data, requiring integrations that send, receive, and sync data from one app to another.



Demo video of interop https://youtu.be/b5EWVgQP3R0?si=DrB9RpH5aau4vGje&t=24

Lix enables more capable apps and services

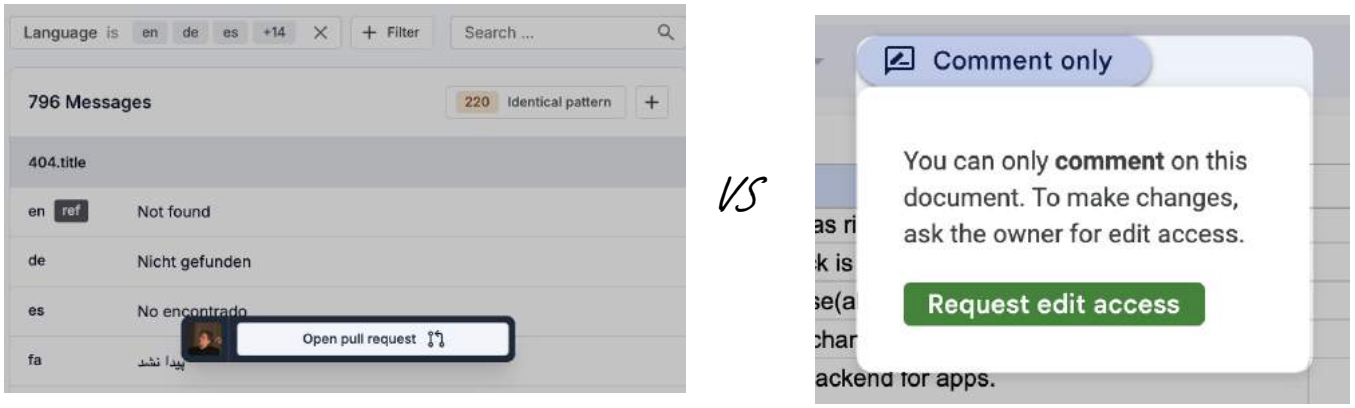
Permissionless collaboration

Lix brings branches and change requests and, thereby, permissionless collaborations to industries other than software engineering. We will see an “open-sourceification” of music, video, architecture, public policy making, fashion, and who knows, maybe even automobiles.

Try it out yourself with Fink https://github.com/samuelstroschein/seed-demo, one of our GTM apps built on lix. See how easily you can contribute translations without asking for permission.

- You need to sign in with GitHub because lix is git compatible and we use GitHub as a host at the moment to spare resources.

- Git, GitHub, and pull requests are hard to understand. We will address the UX challenges with lix and a lix hosting platform in the future.



Lix apps don’t need a “request edit access” workflow, enabling permissionless collaboration.

A large “What if 1000s of artists are able to create a song together?” poster hanging in my apartment as a statement to what lix can enable.

Built-in automation pipelines (CI/CD)

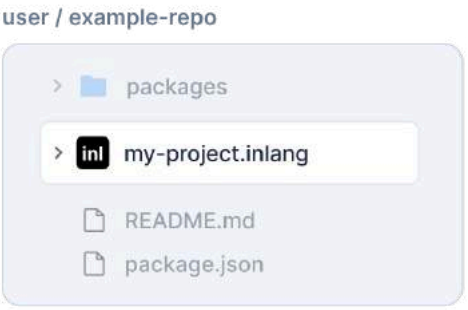
TODO

Lix enables AI

Access to and history of data

A company's data exists as files in a lix repository.

AI services or agents get access to a lix repository and just read (or write) files. No integration, data syncing, or auth complexity. Furthermore, data contained in lix has history, providing LLM's with context. LLM's with context are better than LLM's without context.

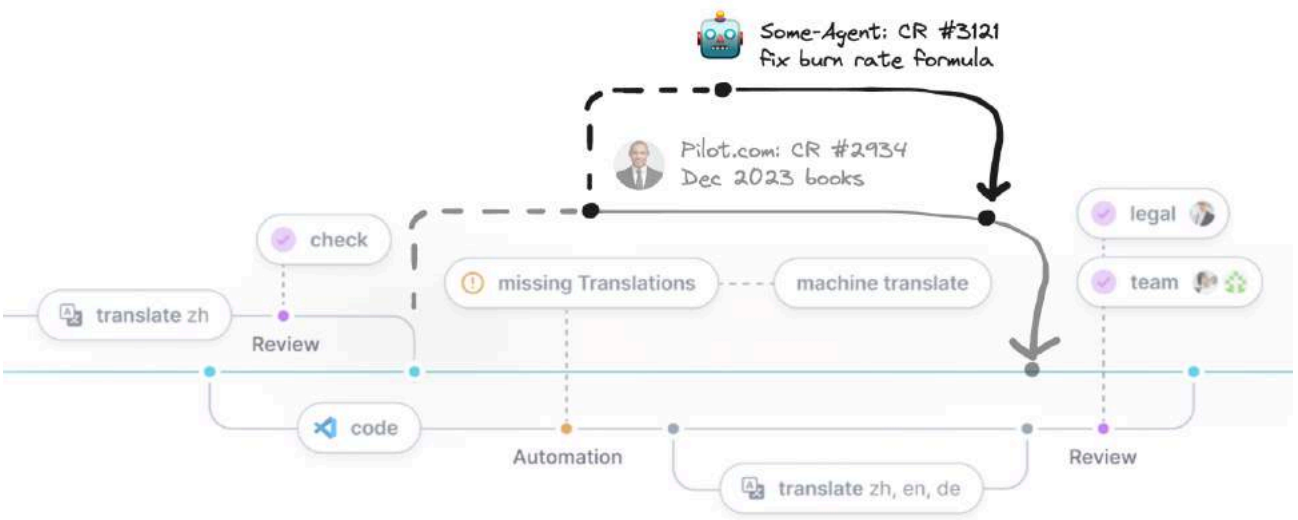


Data is "just" files.

Controllable AI agents

Change control provides a controllable and auditable mechanism for observing an agent's changes and accepting or rejecting them.

Without the ability to control changes of AI agents, AI agents will not get mainstream adoption. Companies need control and audibility over AI agents' work.



A change request from an AI agent fixing a formula in a company's financials.

Cloud-based apps & SaaS will be at risk for disruption

Lix apps provide USPs to customers that cloud-based solutions can't replicate:

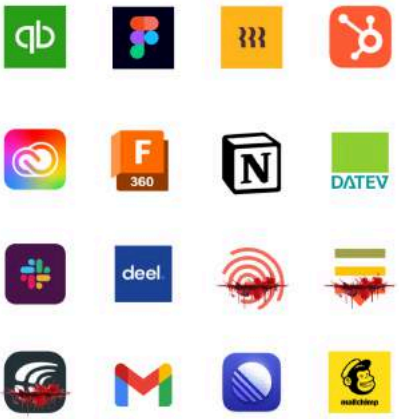
- Data ownership (no lock-in)
- Interop between apps and services
- Change control features like auditing or automation pipelines (CI/CD)

Entrepreneurs will sense the opportunity lix provides to disrupt cloud-based solutions. It's a matter of time. Lix apps being substantially cheaper to develop than cloud-based apps is a reason alone for disruption.

Companies will push to replace cloud-based with lix-based apps. The more lix apps a company adopts, the easier it becomes to collaborate, leverage interop, and build automation pipelines. Cloud-based apps will be of inherent negative value to a company.

Take the following cloud-based apps and ask yourself what value cloud-based provides if collaboration is not a concern:

> **Carta** - Audit proof equity management in the US. What value does Carta provide by owning the cap tables?
>
> **Figma** - Made UI design accessible. What benefit do users have if Figma owns the designs?
>
> **Notion** - Provides a great UX for writing documents. What value does Notion provide by owning the documents?
>
> **Deel** - Deel's makes hiring globally easier. But what value does Deel provide in owning employee data?



The value of cloud-based apps (collaboration) erodes with the emergence of lix. The downsides (expensive to build and maintain, lock-in, and no change control) will lead to the demise of cloud-based apps.

Lix needs a GTM product that yields requirements

Designing and developing lix without a use case would end up in R&D hell, building the wrong things, and getting no adoption.

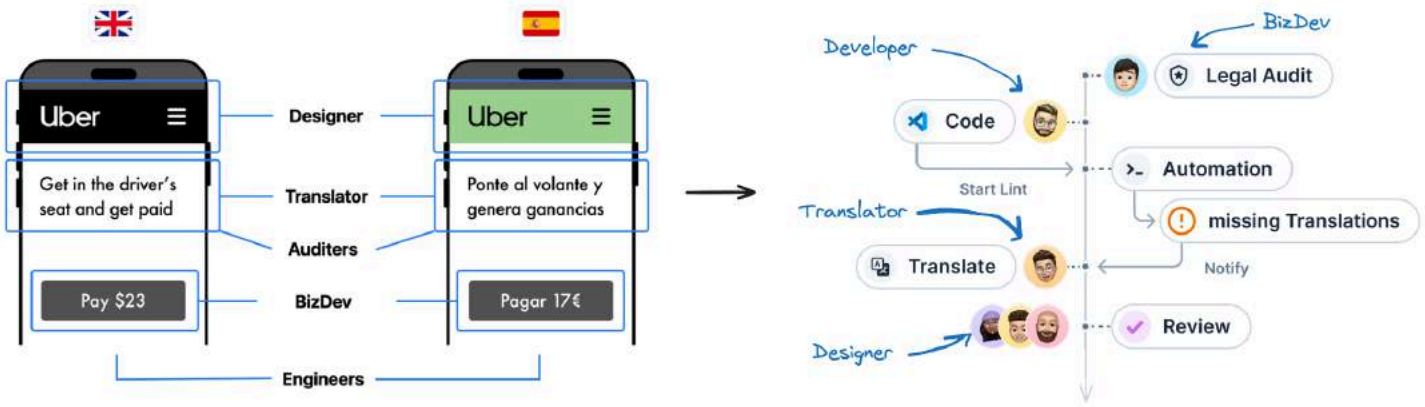
GTM product: Inlang - globalization ecosystem for software

The problem

Globalizing software is a cross-team effort that involves ensuring compliance, adapting the design, handling different currencies, managing translations, writing different marketing copy, etc.

Existing point solutions such as apps to manage translations (Lokalise, Weglot, Smartling), apps to manage design copy (ditto, translator), or i18n libraries for developers (next-intl, i18next) don't have interop between each other.

The lack of interop of existing solutions leads every app and team to have it's own collaboration layer, review system, and automation capabilities. Cross-team collaboration and automation, which is required to globalize software, is hard to impossible.



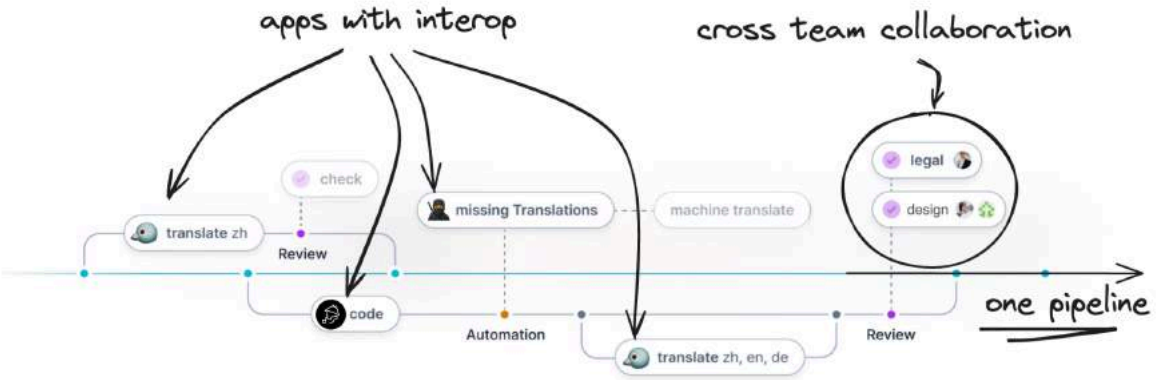
Globalization of software is a process problem that involves multiple teams collaborating in a pipeline.

GTM product: Inlang - globalization ecosystem for software

Solution

Inlang solves software globalization with an ecosystem of interoperable lix apps.

Every app in the inlang ecosystem shares the same review, automation, and collaboration layer, making cross-team collaboration possible. No syncing, hand-offs, or automation barriers.

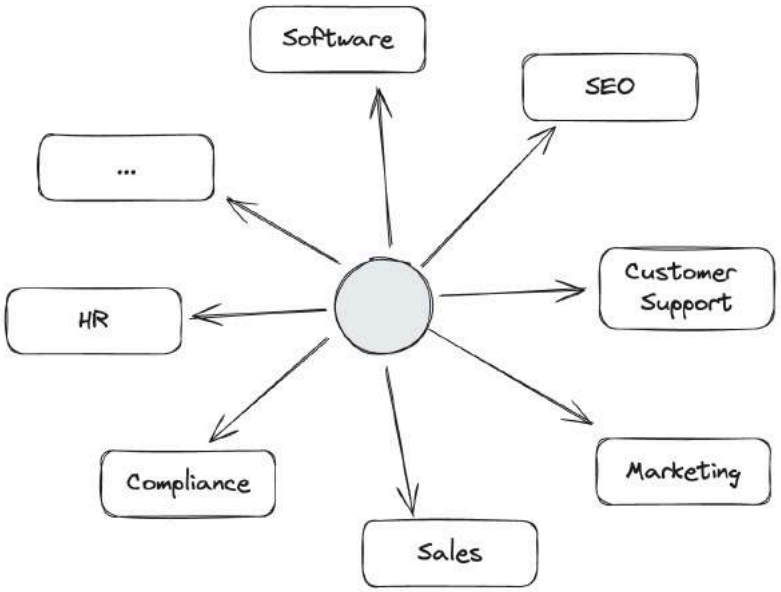


Every app and team collaborates on one change graph, simplifying collaboration and automation of globalization processes.

Future: The globalization shopping mall (Amazon.com for globalization)

Software globalization is the GTM for inlang and opens Pandora's globalization box. Once the product is globalized, the entire company, from support to compliance, needs to be globalized. Even in our early stage, customers already ask us for solutions outside of software globalization.

Expanding inlang from the largest bookstore in the world to the globalization shopping mall will happen :)



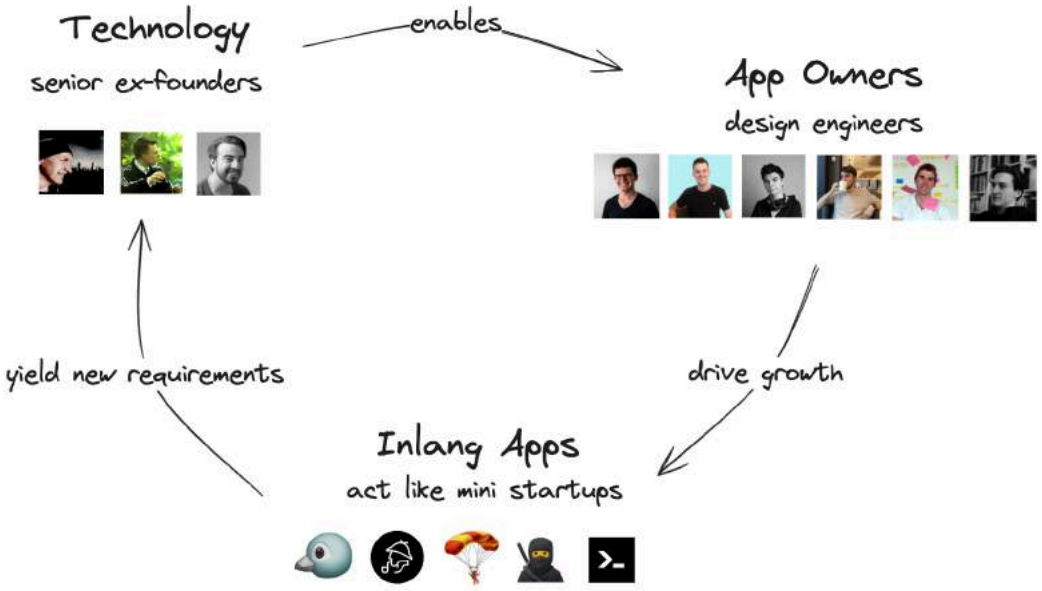
Inlang as the GTM product for lix is why we will win

Inlang yields the requirements for lix

Inlang's multi-app, multi-team, and multi-persona problem yields the right requirements to design lix.

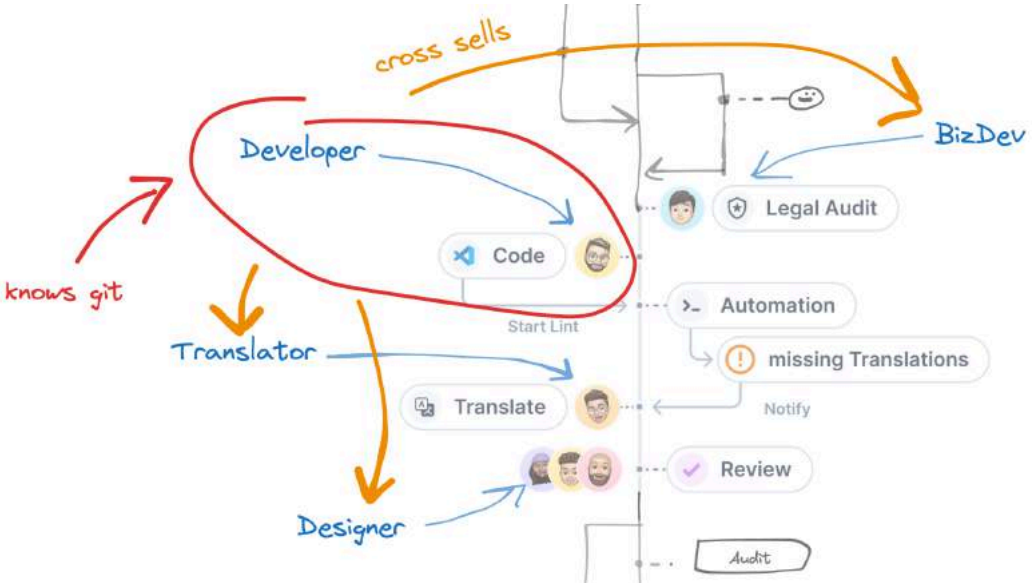
Mini startups that lead to a fast feedback loop

Inlang's multi-use case problem can be split into "mini startups": Every inlang app is a mini startup with high ownership, leading to fast iteration speeds by reducing management overhead.



Cross-selling from developers

Developers understand the value of change control automation pipelines (CI/CD). They adopt inlang apps and cross-sell other personas. We don't have to explain the benefits of lix to non-technical audiences (in the beginning).



Pre-seed achievements

MVP demonstrates that lix is the future

We have a suite of interoperable inlang apps that demonstrate that using lix as the backend for apps is the future:

- The ease of building inlang apps for us is incredible. Load and display a file. Done.
- Out-of-the-box automation pipelines (CI/CD).
- Adoption is substantially easier because apps work locally with no accounts.

MVP raises the expectations of what software must be capable of

Once people understand lix and the benefits of change control, current apps lead to a “would be better with change control” reaction. Entrepreneurs will leverage lix to deliver on those raising expectations of users.

Here is our non-exhaustive list of apps that would be substantially better with lix:

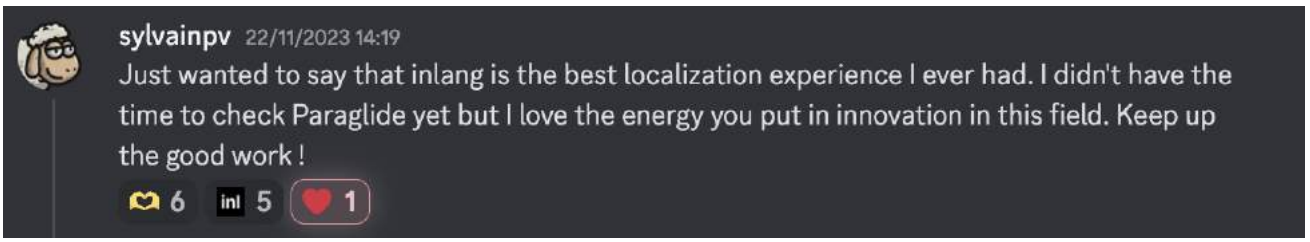
- Product analytics (PostHog, MixPanel)
- Payroll/HR (Deel, Rippling)
- Accounting (QuickBooks, Brex, Ramp)
- Document editors (Google Docs, MS Word, Notion)
- Design software (Figma, Sketch)
- Equity management (Carta, Easop, Pulley)

Pre-seed achievements

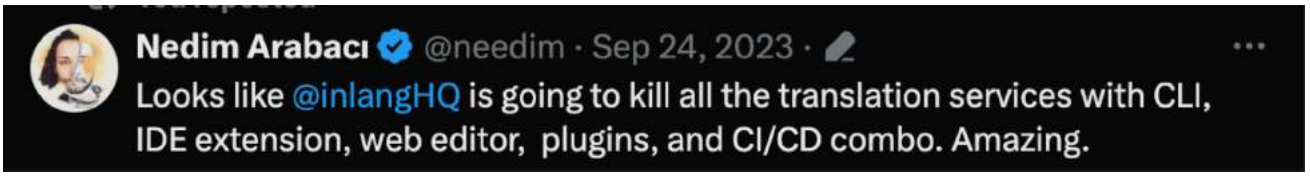
Developers love the MVP despite being extremely limited

Despite having a basic proof of concept that lacks features like gendering or pluralization, users are blown away. While most users don't name lix, lix is the reason we are able to build inlang and exceed user expectations. Branding lix stronger in the future is on the roadmap.

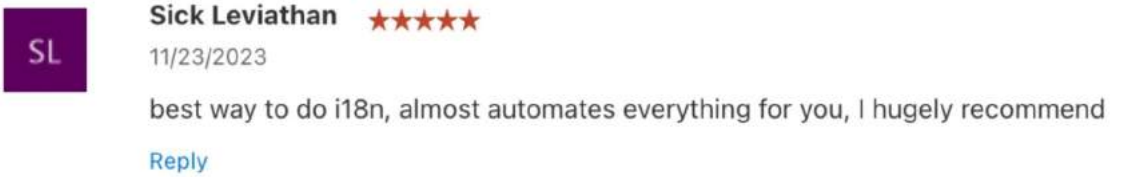
https://dev.to/robertosnap/inlang-paraglidejs-blew-my-mind-3984



https://discord.com/channels/897438559458430986/1096039983116202034/1176965137094148257



https://x.com/needim/status/1706026369677709531



https://marketplace.visualstudio.com/items?itemName=inlang.vs-code-extension

Pre-seed achievements

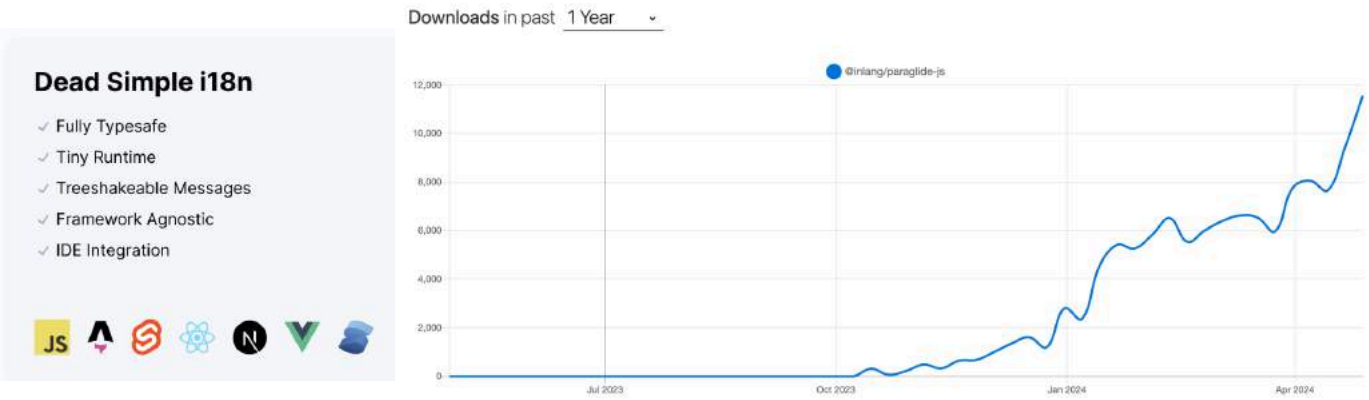
Product-led growth across multiple inlang apps

- Growth is severely limited by building the infra faster and easing the getting started.

- We want to accelerate growth by raising the seed: Build the infra faster, implement cross-selling, and support PLG motion with marketing.

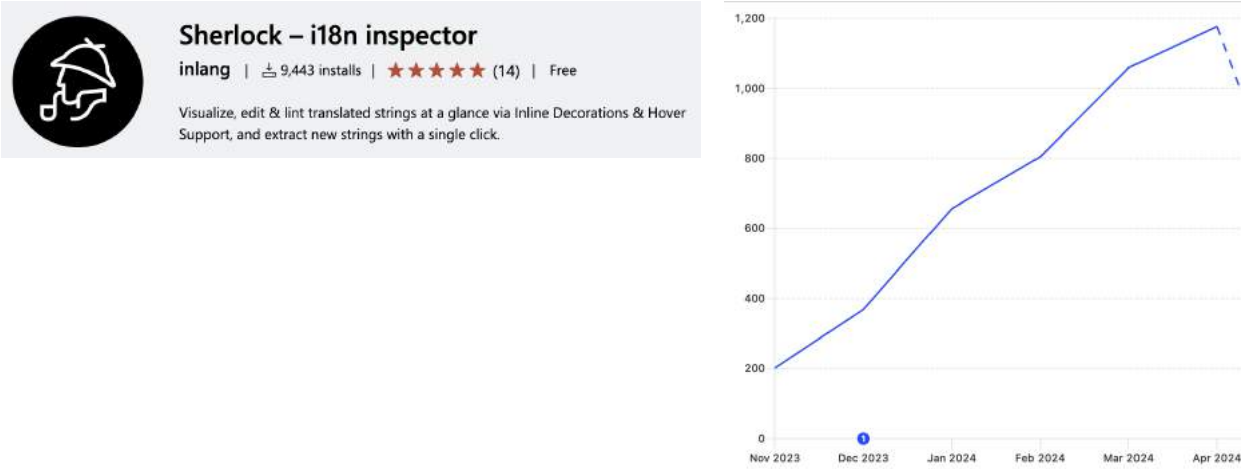
[Paraglide JS]

- is becoming the go-to i18n library for web dev
- launched 4 months ago, already 11k weekly downloads
- growth is limited by building the infra of lix and inlang faster



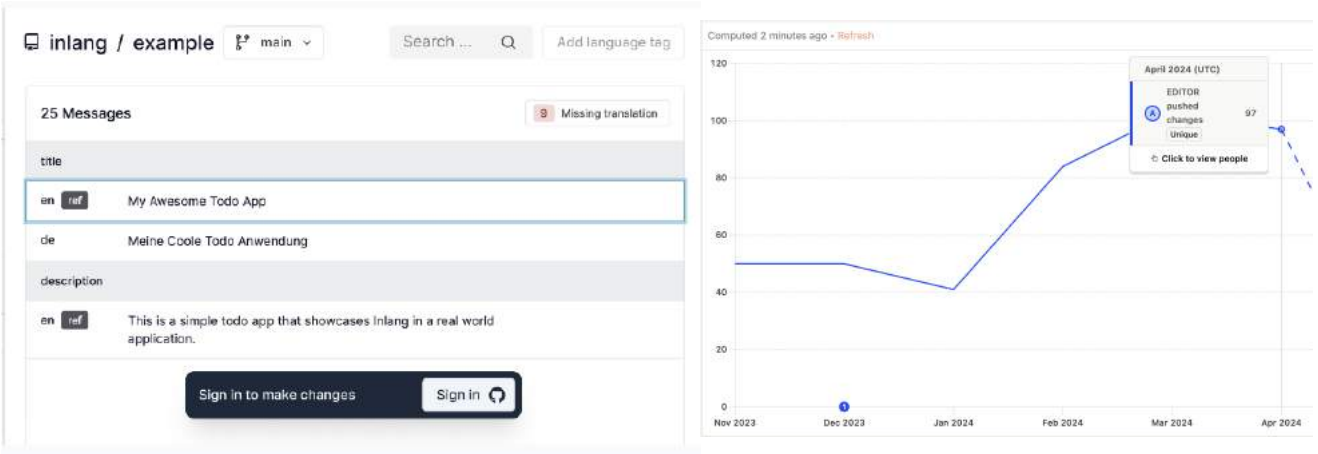
[Sherlock]

- 21.74% average mom growth (last 3 months)
- 9443 installs, 1174 monthly active developers
- Growth is limited by easing the getting started



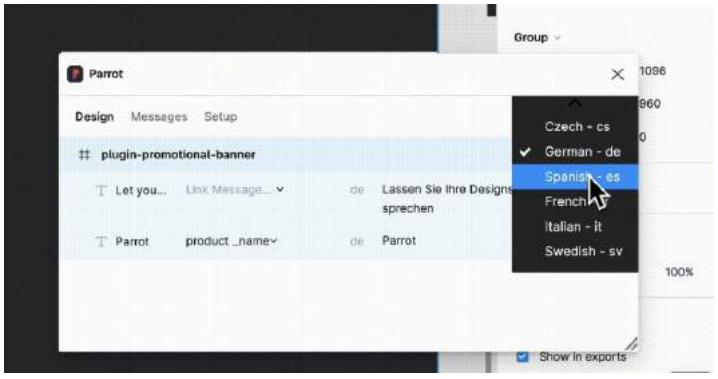
Fink

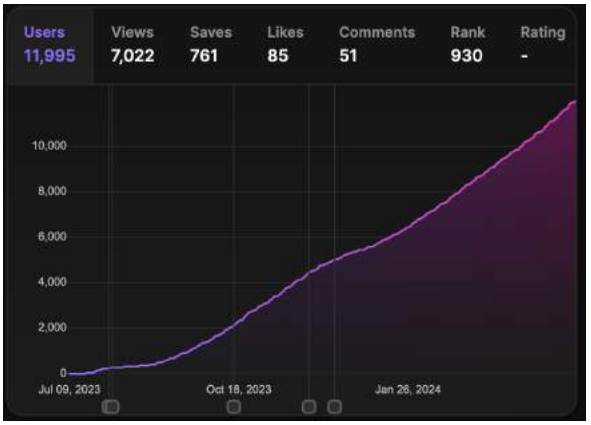
- Roughly 100 translators per month commit and push changes
- aka 100 translators who use git and GitHub (!)
- Growth is severely limited by targeting github as hosting platform. Need our own hosting platform that is easy to use



Parrot

- Is becoming the go-to translation figma plugin for designers
- 12k users, according to Figma
- Growth is limited by connecting Parrot to the rest of the ecosystem (runs locally in Figma atm. Awesome that lix apps are able to run locally!)

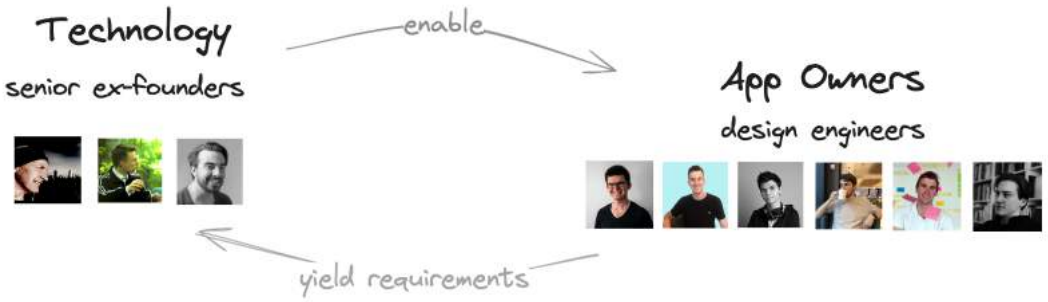




Team

We are the ones to pull this off.

Three ex-founders on the technology stack, design engineers on the app side, and an organization structure that leads to a fast feedback loop.



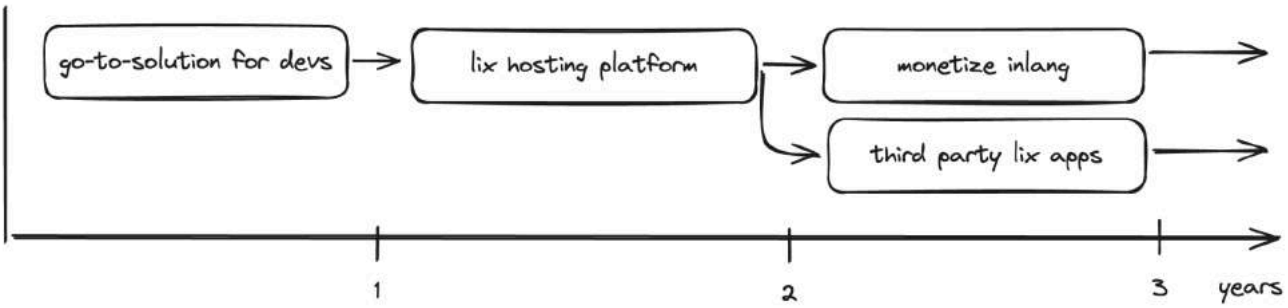
Samuel	CEO Design engineer	Bootstrapped the previous company, experienced the pains of globalization, and founded Opral.
Felix	Design engineer	10 years of web design engineering experience both for fun & for large enterprises. Into Open Source, co-created Drupal admin dashboard.
Niklas	Design engineer	Dual education at Mercedes R&D in sound engineering. Studied with Felix and Nils design afterward. Ofc he is also a DJ!
Nils	Design engineer	5 years of experience in UI/UX and front-end development in both R&D and agency settings. Studied with Felix and Niklas.
Jan	Ex-founder	Over 10 years of experience with data-heavy and sync-enabled web apps and cloud infrastructure. Worked on products with database and file sync at the core. (eg. dropbox clone)
Martin	Ex-founder	Over 15 years of experience in software development and leading teams. Experienced the pains of software globalization firsthand in two organizations (mobile, web, desktop). Deep into local first software and git.
Jürgen	Ex-founder Ex-github	Turned the hiring process upside down by due diligencing the heck out of us, resulting in “this smells like an opportunity.”
Loris	Full-stack Dropped out to join	Got hired because of this.
Jannes	Operations Design engineer	Design engineering background applied to operations.

Roadmap

1. Become the go-to solution among devs with inlang
2. Build the lix hosting platform (“lixhub”)

3.1. Monetize inlang

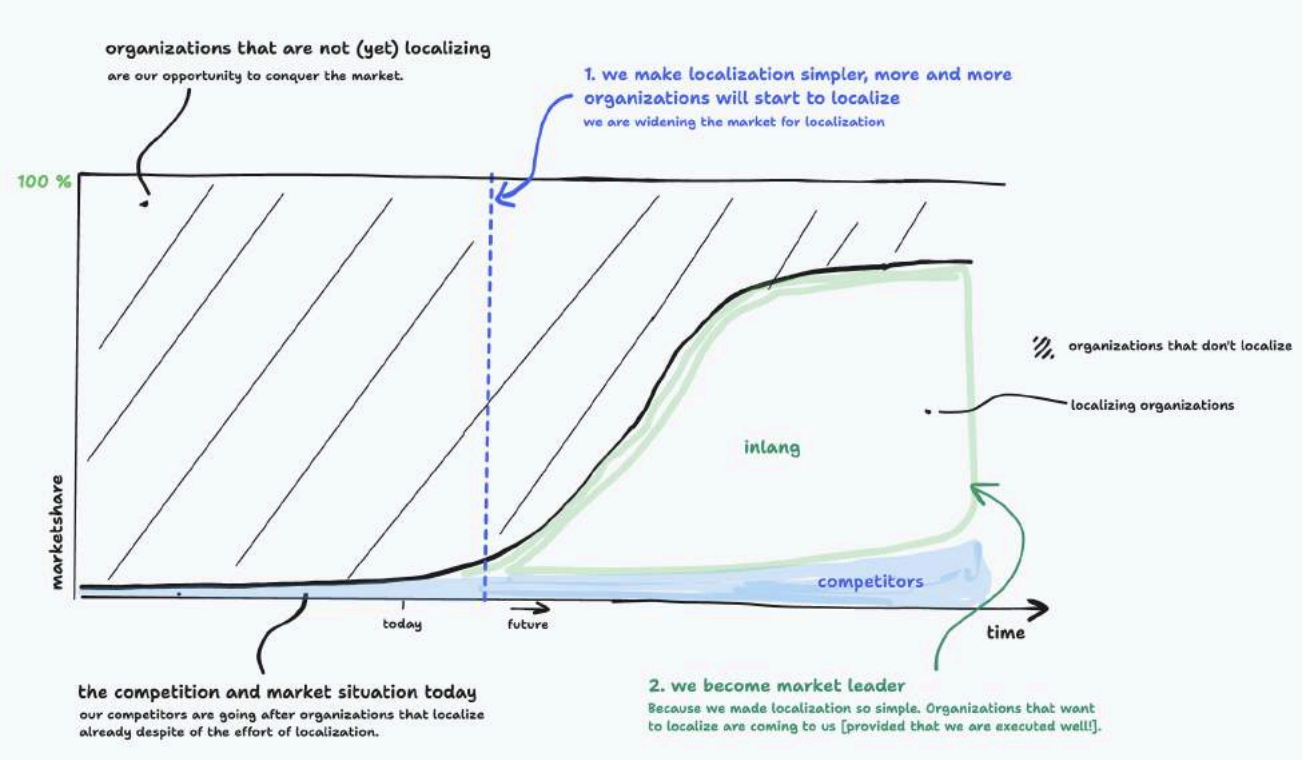
3.2. Third-party lix apps



Appendix

Inlang will widen the market for globalization

The simpler globalization becomes, the more companies will globalize. The more companies globalize, the more competitors must globalize to stay competitive.



Inlang will simplify globalization and widen the (already large!) globalization market.
PS This is a sketch from 2022. Don't get confused by the word "localization".

Future monetization

Both inlang and lix are infra and platform plays. We expect more monetization opportunities to emerge besides the ones we already foresee.

Inlang:

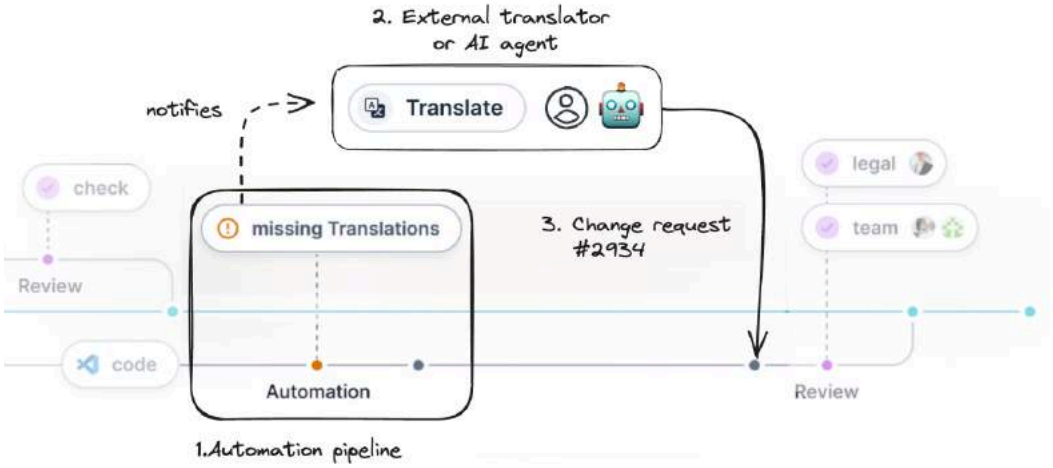
- SaaS for apps
- Globalization as a service marketplace (translators, cultural experts, compliance, etc.)

Lix:

- Lix"Hub" (GitHub for lix)
- App Store (payments, distribution, etc)

Lix enables globalization as a service

Expand to Germany? Subscribe to German translators, design experts, performance marketing specialists, AI agents and cultural experts who help adapt your product one change request at a time to the German, Brazilian, Chinese, US, or French market.



An automation pipeline flags missing translations and notifies an external translator. The external translator, or AI agent, can open a change request that the lix owners can accept or reject.

Carta lix disruption case study

Carta is an example of an app that would be easier to build with lix and provide substantially more value to customers by avoiding lock-in and ensuring interop.

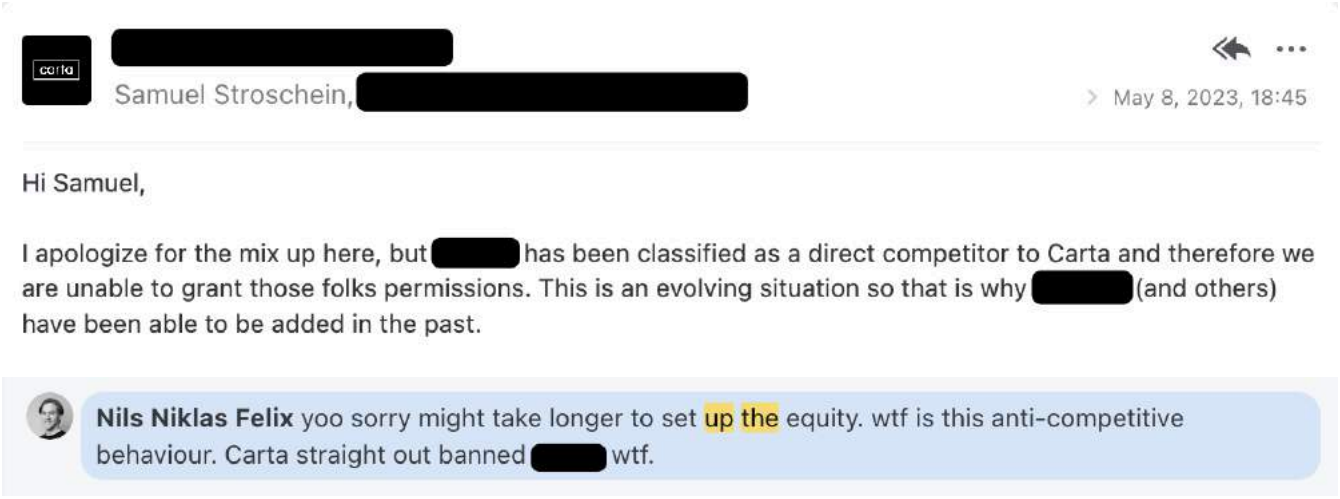
Will a cap table solution built on lix disrupt Carta? We think so. The lock-in Carta achieves leads to unaligned incentives between Carta and their customers. A solution without lock-in and with interoperability is a substantially better solution than Carta. We know it firsthand because we have been burned by Carta's lock-in.

Carta is incentivized to go against customer interests because of lock-in

Carta's ownership of customers' cap tables leads to lock-in that Carta leverages to their benefit and downside of their customers.

Blocking API access and even personal invites (!)

Carta blocked access to our cap table for employees of a solution that makes global equity granting possible. Carta does not support global equity granting (US only). Yet, Carta outright blocked access to employees that we invited to our cap table, leaving us (the customers!) stranded.



Carta sent us an email that they won't grant people we invite access to OUR cap table.

Abusing data ownership with no consequences due to lock-in

The linear founder, Karri Saarinen, called out Carta about abusing their cap table data to promote other business lines of Carta. Unfortunately for Karri, his company is locked into Carta. Carta owns Linear's cap table. Moving to another cap table solution involves effort and risk.

Carta abused customers' data to cross-sell products.
https://twitter.com/karrisaarinen/status/1743398553500971331

Examples of abusive, anti-customer, and anti-competitive behavior are possible because Carta owns its customers' cap tables, creates lock-in, and is incentivized to go against customer interests.

What value do customers get if Carta owns customer's cap tables anyway?

Carta's lock-in is a result of enabling collaboration ("you don't have to send your cap table file around anymore") and providing audit-proof traceability.

However, lix renders both upsides of lock-in obsolete. Collaboration and audit-proof traceability are easier to achieve with lix while avoiding lock-in and ensuring interop for customers.

Carta's remaining value is a UI over "cap table data" and services like 409A valuations. Here again, any cap table app can provide a UI over an open cap table file, potentially much better than Carta's UI. And services are ... just services that need no cloud solution.

Value	Negative value
Traceability for cap table	Lock-in (No ownership over cap table)
Dashboard (collaborative equity management)	No interop with other apps or services
Services (409A valuations, Total Comp)	Incentivized to go against customer interests

A better solution: cap tables as lix files

TODO