

Challenges of building an API service

Learnings from  **svix** and other APIs

Tom Hacoen
Devcon 2022

@TomHacoen
www.svix.com

Who am I?

- Founder and CEO of Svix
- Open source dev and maintainer
- Previously led teams at Samsung and the Israeli intelligence corps



What is Svix?

- Enterprise ready webhooks service
- We help companies send webhooks
- Customers such as: **Brex** and **LTSE**
- Backed by **Y Combinator**, **Aleph**, and amazing founders and CTOs.



What is this talk about?

- What to know before starting an API service
- Learnings from Svix and other services
 - Your requirements may differ
 - Apply reasonable judgement
- Forgot to mention anything? Let me know!



Keep It Simple Silly

- Complexity is your biggest enemy - avoid if possible
- Prevents you from moving fast
- Many points of failure
- Hard to reason about

There is good debt and bad debt

Good debt



Bad debt



Understand your tech stack

- Different technology comes with different trade-offs
- Do you care about consistency? Availability?
- Know the tools you use and their limitations

Know your customers

- What do they want?
- What do they care about?
- What would they hate?
- Have the curiosity of a child.



"If I asked people what they wanted, they would have asked for a faster horse." — Henry Ford



A close-up photograph of a person's hand holding a black power drill. The drill is positioned vertically, with its bit just above a hole in a grey metal rail. The rail is resting on a dark, textured workbench. In the background, a red container filled with drill bits is visible, along with other tools like a wrench and a screwdriver. The scene is dimly lit, focusing attention on the drill and the rail.

It's not what you do,
it's what you enable

A black and white photograph of a boxer in a ring, with a blurred figure of another person in the background.

Understand the solution

- Have a deep understanding of your chosen solution
- But be flexible and dynamic
- *"Everyone has a plan until they get punched in the mouth"* — Mike Tyson

An aerial view of a construction site on a steel deck. Two workers in orange safety gear are visible. One worker is kneeling and using a yellow angle grinder on a steel beam. The other worker is standing and handling a large coil of blue material. The deck is covered with a network of steel beams and vertical rebar. Stacks of steel rods and other construction materials are scattered around.

Set the right foundations



Manage API complexity

- Strive for simplicity, and learn to say no
- Be explicit, watch out for accidental flexibility
- Can't break API - if you build it, you're stuck with it
 - Try to plan for forward compatibility
- Don't leak implementation details

How do you deploy?

- SaaS? On prem? Both? Hybrid?
- Are you limited to AWS? GCP? Only PostgreSQL?
- Have any other runtime dependencies?

The background image is a high-angle, wide shot of a large, modern library atrium. The space is characterized by multiple levels connected by wide, light-colored stairs. The walls and upper levels are lined with floor-to-ceiling bookshelves filled with books. The lighting is soft and even, creating a calm and scholarly atmosphere. The overall color palette is muted, with greys, whites, and the warm tones of the book spines.

Choose your data layer wisely

"Bad programmers worry about the code. Good programmers worry about data structures and their relationships." — Linus Torvalds

A photograph of a goat with dark brown and black fur, standing in a dry, dusty environment. The goat is tied to a wooden post with a thick rope around its neck and another around its middle. The background is a hazy, dry landscape with some sparse vegetation.

People will still use it wrong

- Make it easy to use right
- Make it hard to use wrong
- Expect it to be used wrong and be ready

Find your north star

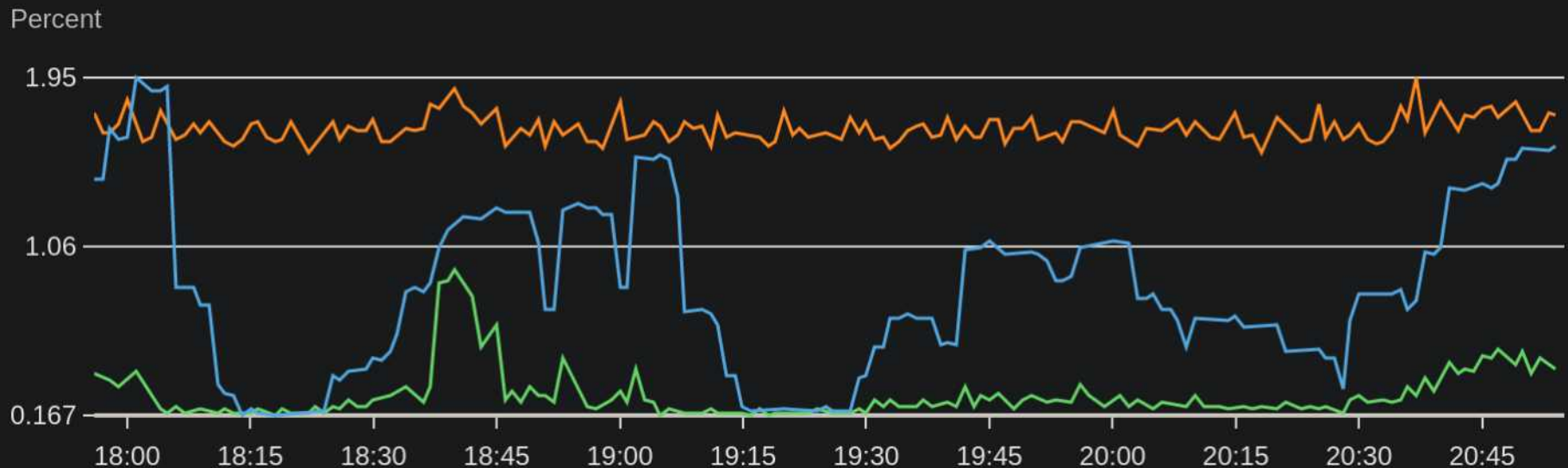
- For us it's RELIABILITY, so:
 - Never lose a message once accepted
 - Avoid downtime at all costs
 - High speed and low latency

An aerial photograph of a large cable-stayed bridge spanning a wide body of water. The bridge features two prominent piers with multiple stay cables supporting the deck. The water is dark, and a small boat is visible in the lower center. The background shows a distant shoreline with some buildings and greenery. The text "Be dependable" is overlaid in white, bold, sans-serif font in the center of the image.

Be dependable

Monitor everything

Have fully visibility into your systems



Logs everywhere

- You want to be able to detect anomalies
- Have information when something goes wrong
- Watch out not to log:
 - PII & PHI
 - Sensitive information

Reliability

- Have redundancies in place
- No single point of failure
- Test your systems under load
- Seek out failure cases and address



Ensure high availability

If you are down, your customers are down

Accountability

- Have a public status page
- Be quick and transparent about failures
- Reach out to affected parties immediately
- Do post-mortems to understand failures
- Hold yourself to the highest standards

Define and test behaviors

- Test all routes and behaviors
- Use regression tests to lock behavior
- Ensure high code coverage (though not enough!)
- Test changes and have ongoing tests
- We encode constraints in the type system —
thank you Rust!

A dark, grainy photograph of a coastal town, likely after a disaster. In the foreground, a boat is partially visible, with the name 'YUKON' on its side. The town is built on a hillside, with many houses and buildings. The overall tone is somber and desolate.

Disaster recovery

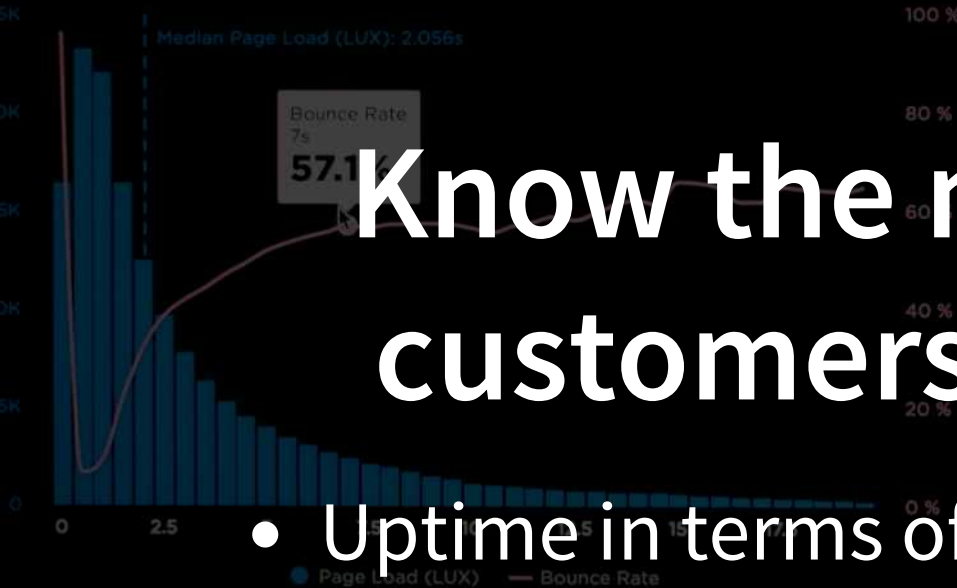
- Hope for the best, prepare for the worst
- Have off-site backups and point-in-time recovery
- List what can go wrong and how to address
- Train staff on recovering from failures

Know your dependencies

- You're only as strong as your weakest link
- Only use trusted tools from trusted sources
- Make sure your tools are dependable
- Evaluate your dependencies yourself
- It's OK to be bold, if you are also prudent

USERS: LAST 7 DAYS USING MEDIAN ▾

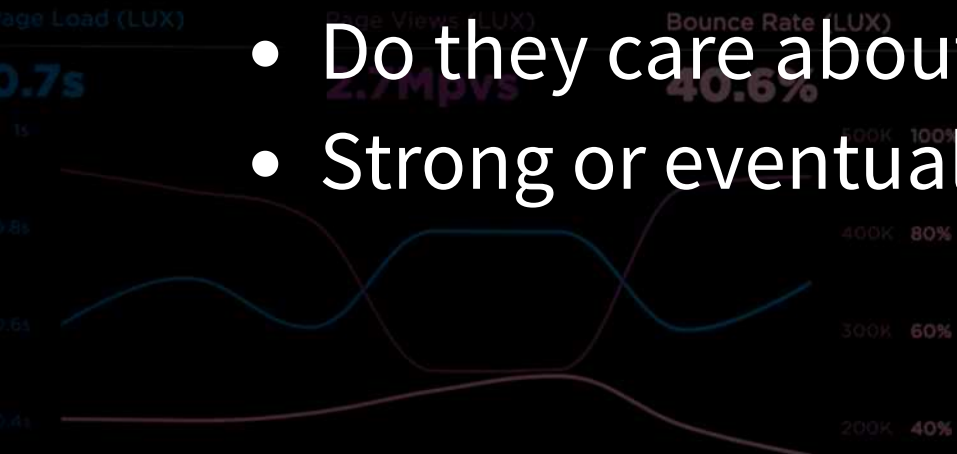
LOAD TIME VS BOUNCE RATE



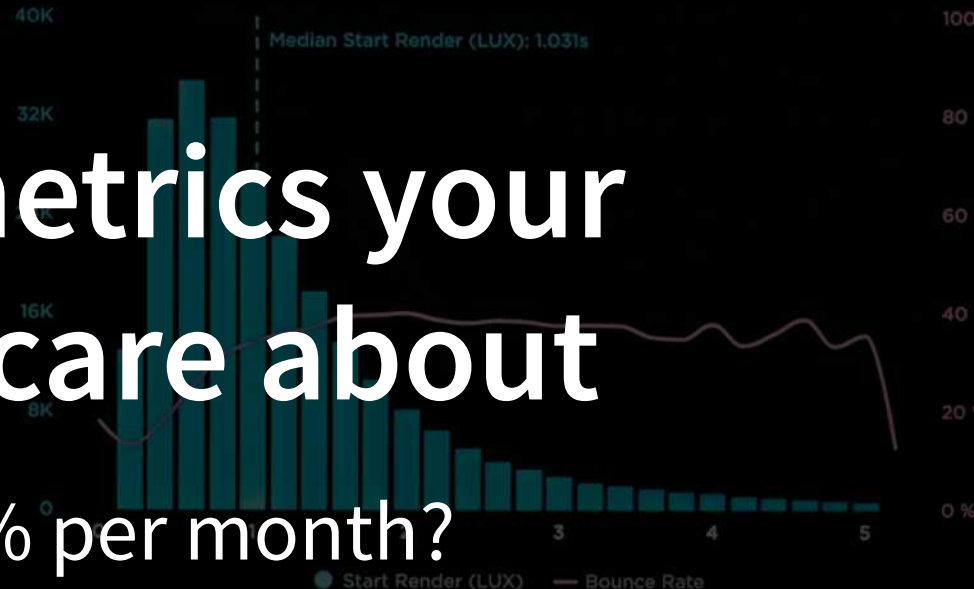
Know the metrics your customers care about

- Uptime in terms of % per month?
- Uptime in terms of # of API errors?
- Do they care about latency? Throughput?
- Strong or eventual consistency?

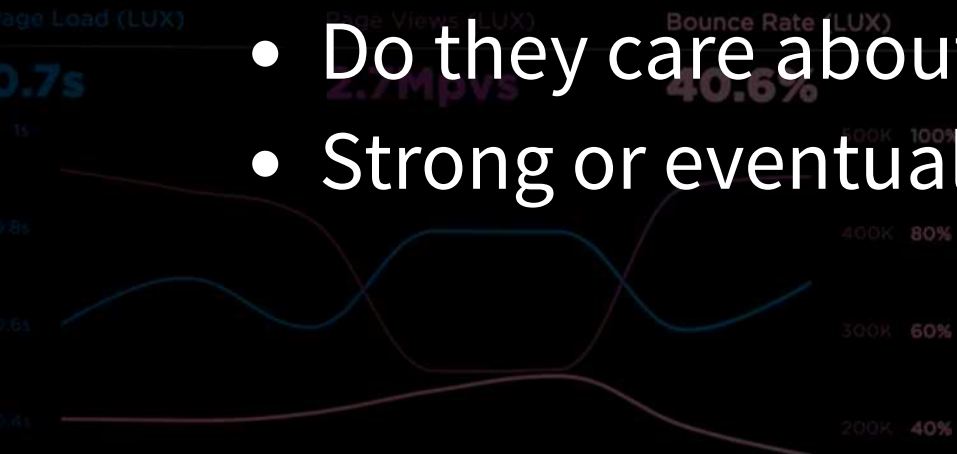
PAGE VIEWS VS ONLOAD



START RENDER VS BOUNCE RATE



PAGE VIEWS VS ONLOAD



SESSIONS





Application security

- OWASP Top 10 and best practices
- Correct password handling
- Strict authorization and restrictions



Operational security

- Vulnerability handling - patching systems
- Security and dependency scanners
- Security training for the team
- Encryption in transit and at rest

Strict and locked down

- Least privilege access control
- No one should have direct access to prod
- Limit capabilities of accounts with access
- Paper trail - log all access and operations



Isolating data between customers

- Multi tenant? Single tenant?
- Can one customer access another's data?
- Have strong enforcement for that



Make security easy for your customers

- Role based access control
- Enable key rotation and scoped keys
- Educate them about security with your service
- Make it hard to get security wrong



Educate yourself with security
best practices



It's more than just code

- There is a lot of devops
- There is a lot of infrastructure
- Use infra-as-code - don't touch the UI!
- Build or buy? I prefer buy



**Prepare for the future,
but build for the present**

A large yellow construction crane is the central focus, its lattice boom extending diagonally across the frame from the bottom left towards the top right. The crane is set against a dark, overcast sky. In the background, other parts of construction cranes are visible, including a smaller one on the left and another boom on the right. The overall tone is industrial and somewhat somber due to the dark lighting.

Make it scalable, but not too much

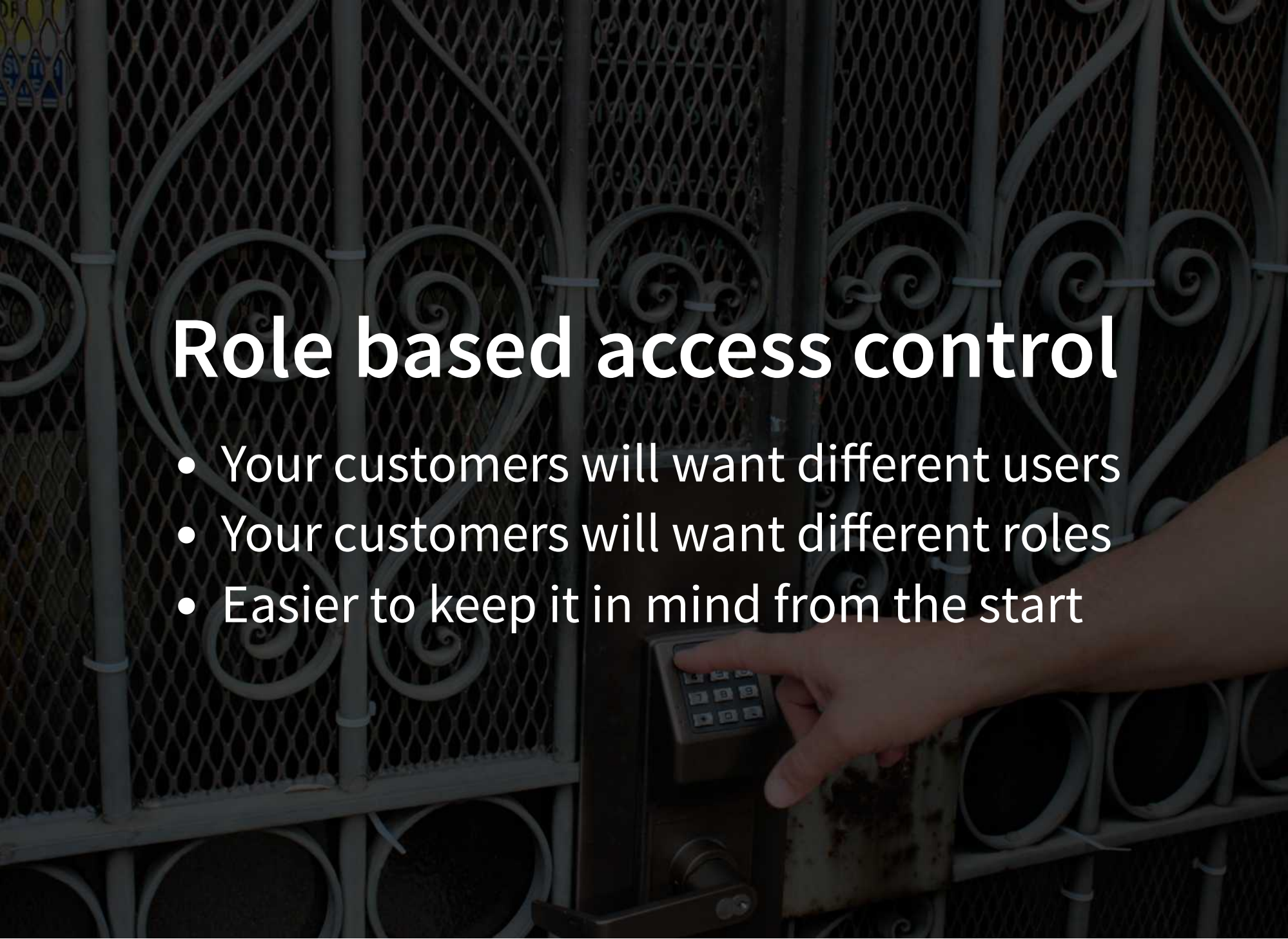
- Build for 2-5x your current expected scale
- Have a path in mind for 10-20x

An aerial photograph of a dense city skyline, likely New York City, taken at dusk. The sky is a deep blue with scattered clouds. The city is filled with numerous skyscrapers and buildings, many of which are illuminated. The text "Multi region, multi environment" is overlaid in the center in a white, sans-serif font.

Multi region,
multi environment

Role based access control

- Your customers will want different users
- Your customers will want different roles
- Easier to keep it in mind from the start



Be flexible and extensible

- Your customers are developers
 - They may want to use it differently
 - Let them build cool things on top of you
- Send webhooks (using Svix...)
 - So they can build automation on top

The devtool equivalent of UX





Be consistent

API Standards vs. BDFL

Great docs make the difference

- Both overview and deep-dive
- Both beginners and advance
- Show how your API should be used
- Don't assume people know your product
 - The curse of knowledge

Don't forget about tutorials

- Guide developers through common uses
- Get them started quickly - easy onboarding
- Highlight cool features

Even developers need support

- Maybe your API isn't clear
- Maybe they found a bug or want a feature
- Maybe they just need help
- Make it easy to debug and report issues

Have a great API

Good

```
await fetch('https://api.svix.com/api/v1/app/', {  
  method: 'POST',  
  headers: {  
    'Content-Type': 'application/json'  
    'Accept': 'application/json'  
    'Authorization': 'Bearer ' + token,  
  },  
  body: '{"name": "some name", "uid": "app_234"}',  
});
```

Have an awesome SDK

Better

```
const svix = new Svix(token);  
  
await svix.application.create({  
  name: "some name",  
  uid: "app_234",  
});
```



BONUS:

Would like to offer on-prem?

- It's A LOT of additional work
- Support multiple environments and way to deploy
- You need more configuration options
- Even more documentation
- You have to think about data migrations
 - You can no longer manage that

A silhouette of a person standing in the center, looking out through a large, arched opening. The person is holding a suitcase. The background is a dramatic sunset or sunrise sky with clouds, and a body of water or a path leads out from the archway. The overall mood is contemplative and hopeful.

Closing words

- Main takeaway: it's not all about the code
- These worked for us, you may be different
- Something missing? Tweet at [@TomHacohen](https://twitter.com/TomHacohen)

Questions?



Website: www.svix.com

Twitter: [@SvixHQ](https://twitter.com/SvixHQ)

Github: github.com/svix/svix-webhooks