Good enough engineering to start an Internet company

By **Wenbin Fang** · March 11, 2019

SHARE

I gave a guest lecture in an undergraduate software engineering class (CSCE431) at Texas A&M University a few days ago. Now I've turned this lecture into a blog post here, and hopefully some people on the Internet will find this useful.

If you arrived from a Google search, here are some contexts:

I'm running a small Internet company — Listen Notes, Inc. — with only one full-time employee (me), as of March 10, 2019. We built a podcast search engine website ListenNotes.com and a podcast API.

Good enough engineering to start an internet company

Wenbin Fang

LISTEN NOTES

wenbin@listennotes.com

I'll share with you my experience about starting an internet company. Building an internet product is not like building an iPhone or a pyramid. Your product doesn't need to be perfect at the beginning. If you are building something useful, other people will tell you what to do next. And you'll figure out what's next. Generally, you should be comfortable dealing with uncertainty, if you want to start your own company.

The first version of Facebook was launched in early February 2004, which was an undergraduate student's mere four weeks worth of work. It was a good enough product with good enough engineering. Every Computer Science college graduate nowadays should be able to build the first version of Facebook over a weekend, using a modern web programming framework (e.g., Rails, Django...).

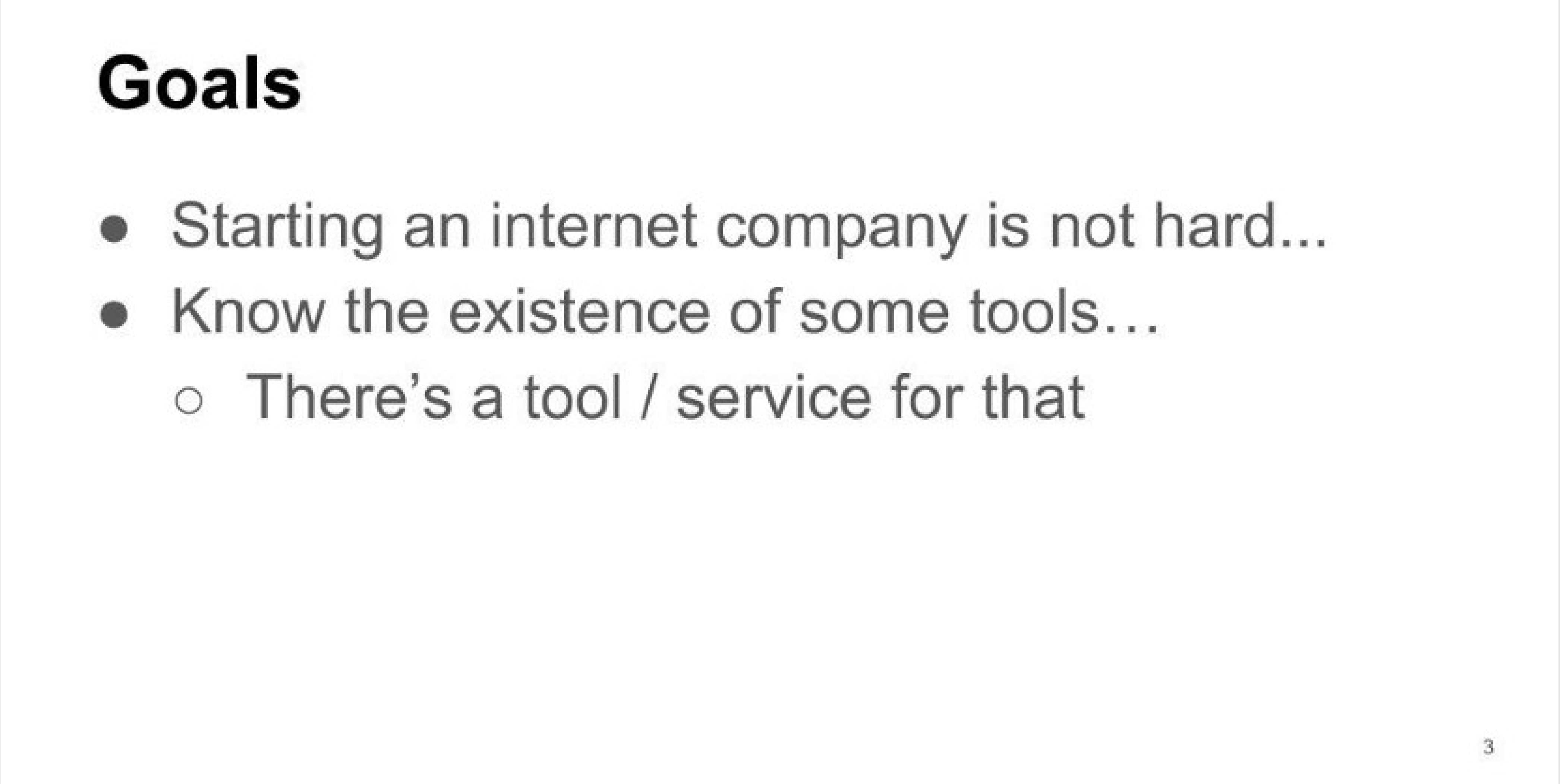
Internet companies

Google
amazon
facebook
Dropbox
LinkedIn
stripe
Uber
SnapChat
airbnb
lyft
Spotify
DISCORD
twitch
slack
Pinterest

2

Companies such as Google, Snapchat, Spotify, Amazon, Twitter, and others are all great internet companies in our generation.

However, I can't tell you how to become a company as successful as them. I'm not there yet. These companies are so successful and so big. For example, Google has 100,000 employees, as of March 2019. But there was a moment in the history of time when Google had only two employees. You have to start from somewhere or nowhere. I can talk about how to get started.



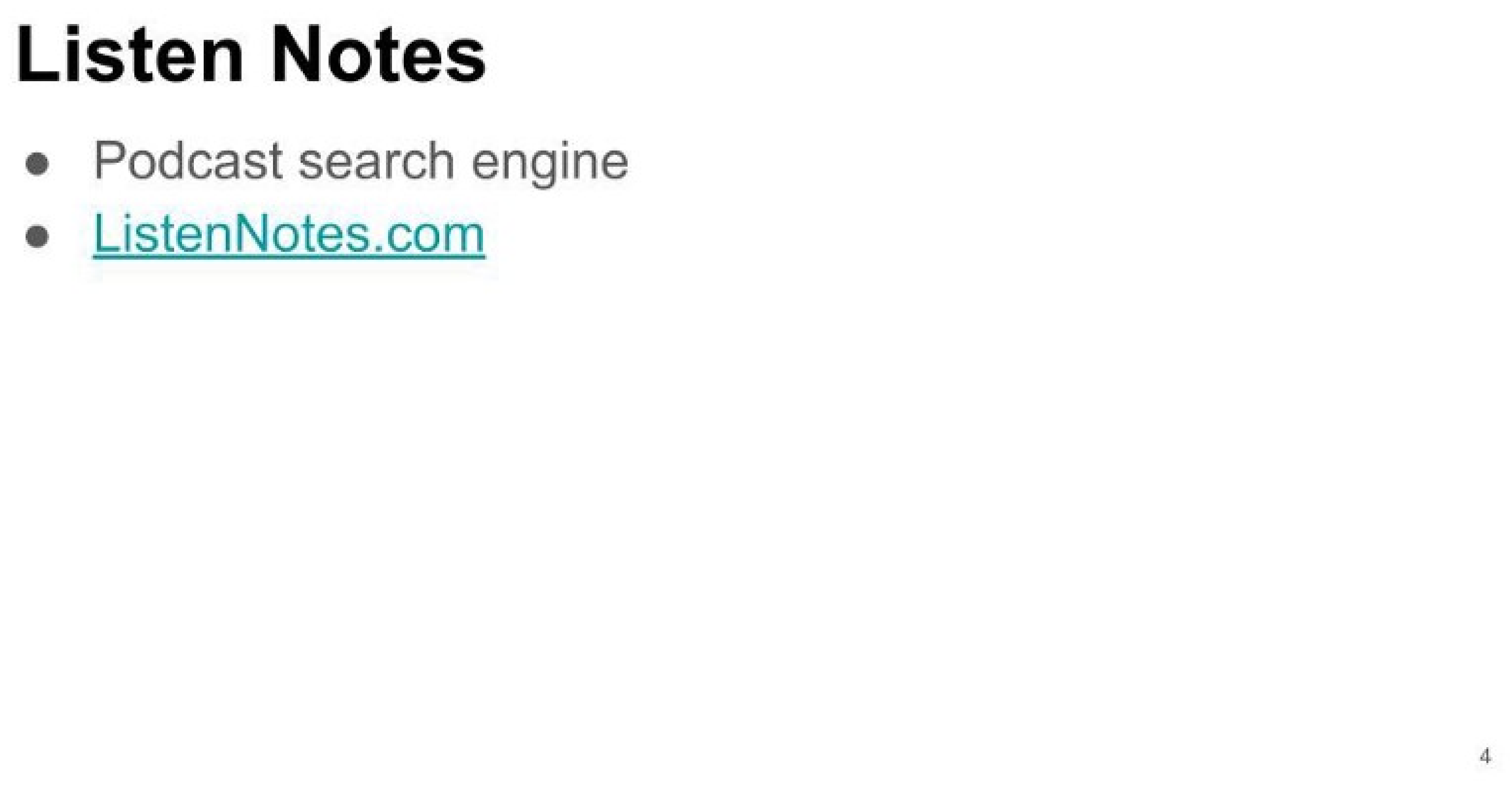
Goals

- Starting an internet company is not hard...
- Know the existence of some tools…
  - There's a tool / service for that

3

Hopefully after this lesson, you'll feel that starting an internet company is not hard at all. And you'll learn about some tools and services that you can use in your future projects.

You'll hear me saying "there's a tool" a lot. This is what I mean by "good enough engineering." It's 2019 now. It's unlikely that you are the first person to encounter a fundamentally new problem. There must be tools and services out there that can help you solve problems—oftentimes, for free!



Listen Notes

- Podcast search engine
- ListenNotes.com

4

Before we delve further, let me provide a brief introduction of Listen Notes.

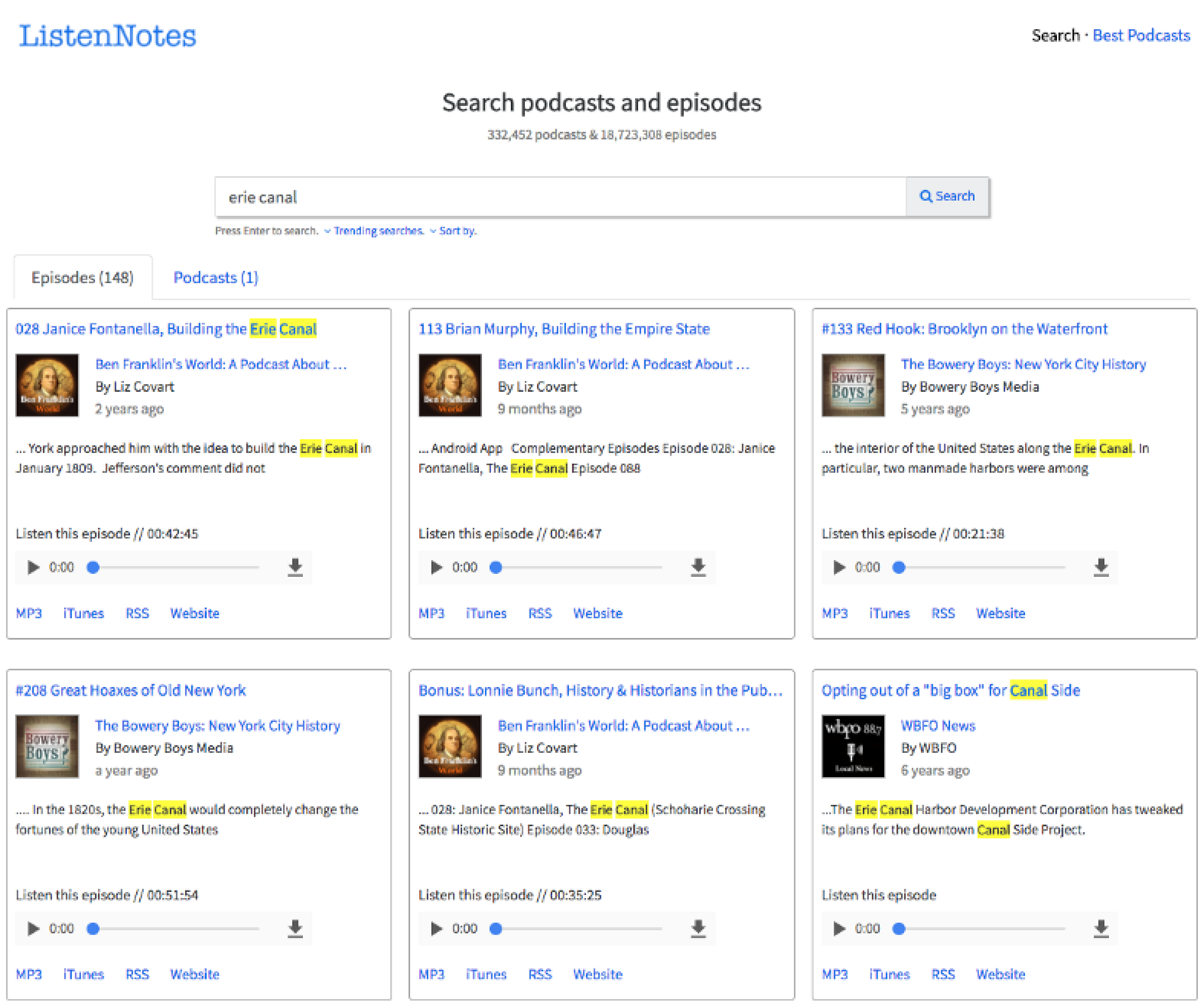
Listen Notes is a podcast search engine website. You type in a keyword and you search the whole internet's podcasts. It's as simple as Google :) But simple doesn't mean easy. We also built a lot of tools on top of the core search engine to help people discover & enjoy podcasts.

Check out Listen Notes: https://www.listennotes.com/

Let's take a step back to talk about podcasts. A podcast is a type of media format. Some people call it "on-demand radio." We consume tons of media content everyday. We watch YouTube videos, watch TVs, read books, listen to music, and listen to podcasts. We consume media content because we want to get information, gain knowledge, and be entertained. Nowadays you can literally learn any topics by listening to podcasts. You can listen to podcasts while your eyes & hands are busy (e.g., driving, working out, walking, doing housework…).

A little more than two years ago (early 2017), I found myself consuming more information from podcasts than from other media formats (e.g., TVs, books…). I needed a podcast search engine that I could use to search & find individual episodes to binge listen to. In hindsight, a podcast search engine should be a very straightforward thing to exist. But I couldn't find a good one. So I spent less than one week building a prototype of Listen Notes, the podcast search engine birthed out of my own wishes and necessities.

I launched the prototype and used it a lot myself. But I didn't touch the code for about nine months, until I decided to work on it full-time after I left my first failed startup. That was September 2017. Cut to 1.5 years later, I'm still having fun working on Listen Notes :)



ListenNotes

Search · Best Podcasts

Search podcasts and episodes

332,452 podcasts & 18,723,308 episodes

erie canal

Search

Press Enter to search. Trending searches. Sort by.

Episodes (148) Podcasts (1)

028 Janice Fontanella, Building the Erie Canal

Ben Franklin's World: A Podcast About …
By Liz Covart
2 years ago

… York approached him with the idea to build the Erie Canal in January 1809. Jefferson's comment did not

Listen this episode // 00:42:45

0:00

MP3 iTunes RSS Website

113 Brian Murphy, Building the Empire State

Ben Franklin's World: A Podcast About …
By Liz Covart
9 months ago

… Android App Complementary Episodes Episode 028: Janice Fontanella, The Erie Canal Episode 088

Listen this episode // 00:46:47

0:00

MP3 iTunes RSS Website

#133 Red Hook: Brooklyn on the Waterfront

The Bowery Boys: New York City History
By Bowery Boys Media
5 years ago

… the interior of the United States along the Erie Canal. In particular, two manmade harbors were among

Listen this episode // 00:21:38

0:00

MP3 iTunes RSS Website

#208 Great Hoaxes of Old New York

The Bowery Boys: New York City History
By Bowery Boys Media
a year ago

…. In the 1820s, the Erie Canal would completely change the fortunes of the young United States

Listen this episode // 00:51:54

0:00

MP3 iTunes RSS Website

Bonus: Lonnie Bunch, History & Historians in the Pub…

Ben Franklin's World: A Podcast About …
By Liz Covart
9 months ago

… 028: Janice Fontanella, The Erie Canal (Schoharie Crossing State Historic Site) Episode 033: Douglas

Listen this episode // 00:35:25

0:00

MP3 iTunes RSS Website

Opting out of a "big box" for Canal Side

WBFO News
By WBFO
6 years ago

…The Erie Canal Harbor Development Corporation has tweaked its plans for the downtown Canal Side Project.

Listen this episode

0:00

MP3 iTunes RSS Website

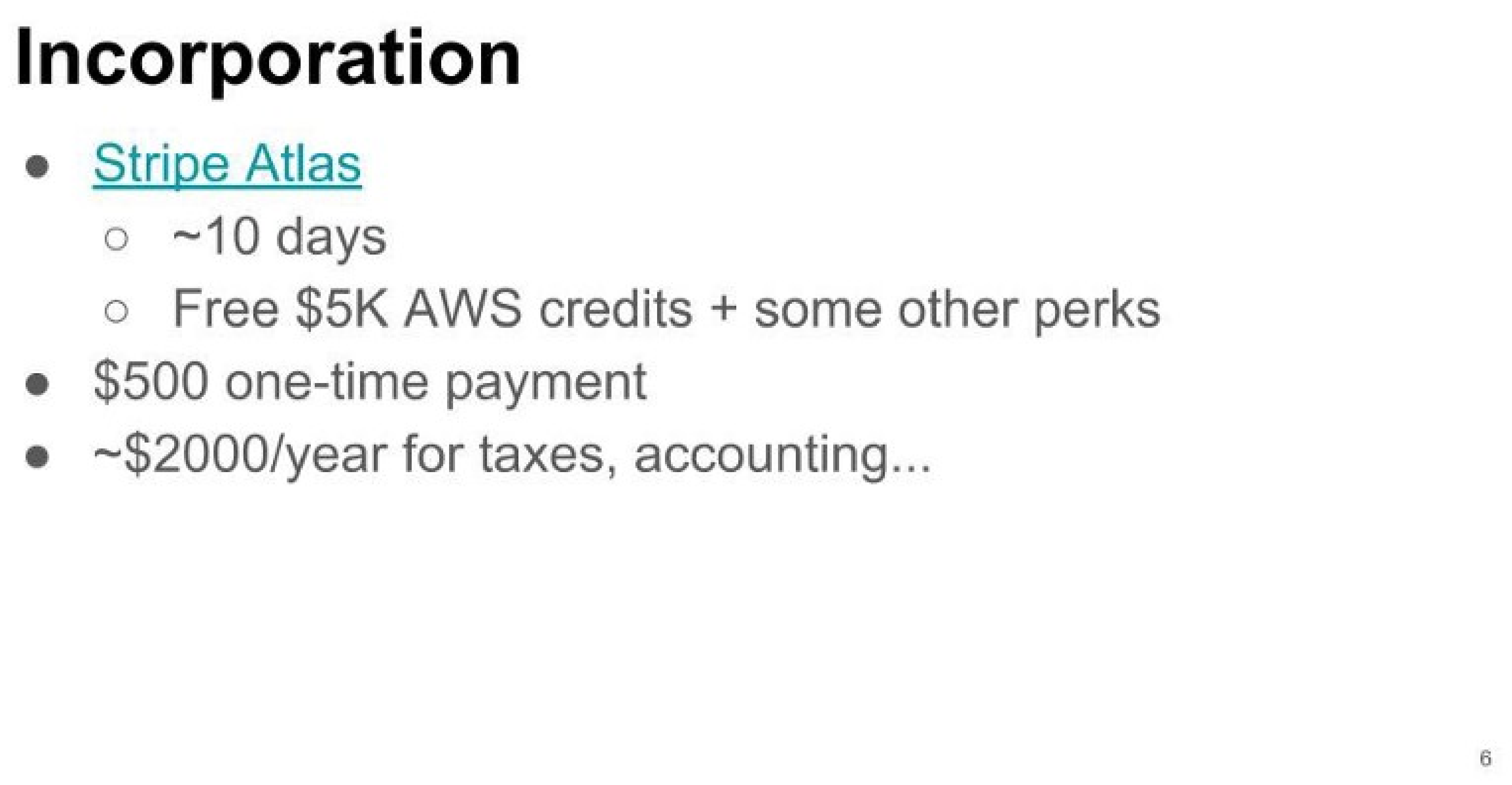
Very first version of Listen Notes. Credits: Lifehacker

Office

- WeWork

Do you need to know where to find an office? Yes, there's a service for that. I use WeWork. I've got a small office inside WeWork in San Francisco. The office is not cheap, but I think it's a good investment for productivity.

I can choose to spend money being 200% more productive myself, or spend money hiring one more employee. I can choose to save money and waste more time, or to save time and make more money. If you were me, what would your choice be?



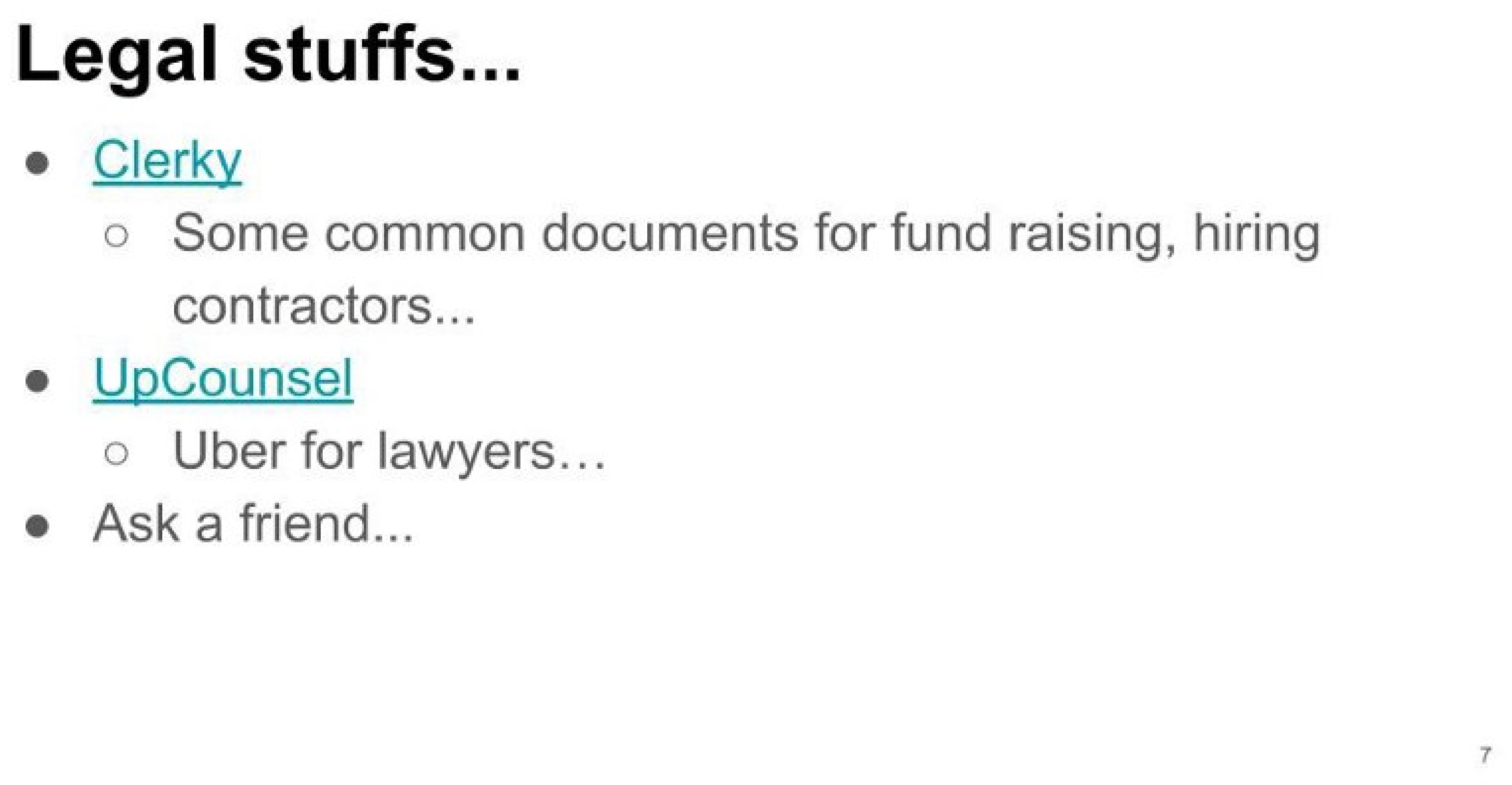
Incorporation

- Stripe Atlas
  - ~10 days
  - Free $5K AWS credits + some other perks
- $500 one-time payment
- ~$2000/year for taxes, accounting...

6

By starting an internet business, we need to have a formal company, a legal entity. There's a service for that.

I used Stripe Atlas to incorporate Listen Notes, Inc. I paid $500, and I got a Delaware C Corp within 10 days. Stripe Atlas provides some nice perks, e.g., $5k AWS (Amazon Web Services) credits. But to keep the company around, I have to pay ~$2k/year for taxes, accounting, and other stuff. This gives you a basic idea of the minimum cost of running a company.



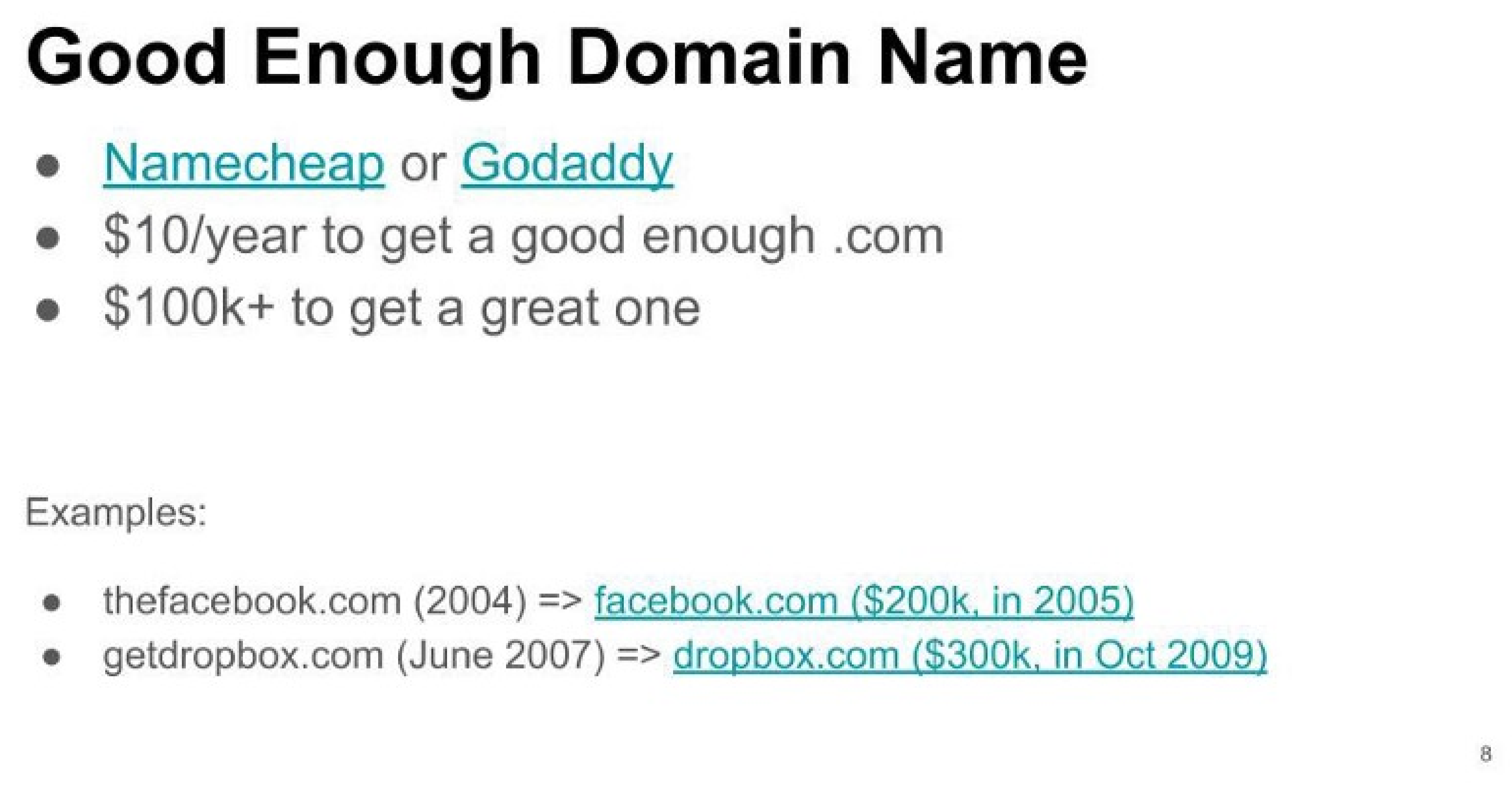
Legal stuffs...

- Clerky
  - Some common documents for fund raising, hiring contractors...
- UpCounsel
  - Uber for lawyers…
- Ask a friend...

7

Because we are a company now, we have to deal with some legal stuff.. There's a service for that.

You can use Clerky to generate legal documents or use UpCounsel to get a lawyer. I've used both services. They aren't perfect, but they worth the money. You can't expect to get Ritz-Carlton-level services with a McDonald's price, right? If you want to be happy, you'd better set the correct level of expectation.



Good Enough Domain Name

- Namecheap or Godaddy
- $10/year to get a good enough .com
- $100k+ to get a great one

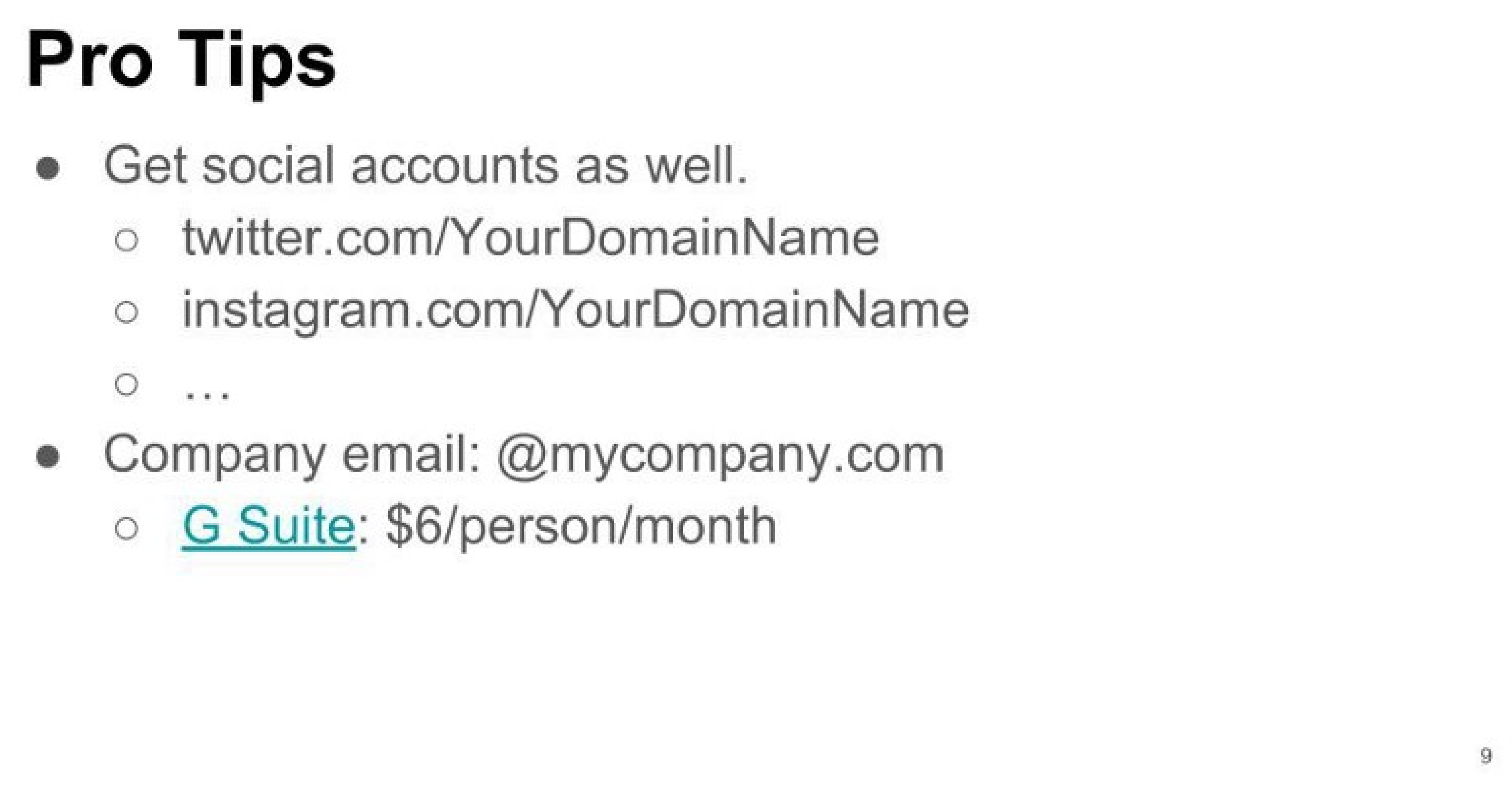
Examples:

- thefacebook.com (2004) => facebook.com ($200k, in 2005)
- getdropbox.com (June 2007) => dropbox.com ($300k, in Oct 2009)

8

To get started, you just need a good enough domain name—a $10/year .com domain. You can always buy a great domain name later.

For example, Dropbox used getdropbox.com for more than 2 years before they bought dropbox.com for $300k. How did I discover that kind of trivia? I listen to podcasts! There was a podcast interview of Drew Houston (Dropbox co-founder & CEO), where he talked about how they secured the dropbox.com domain name.



Pro Tips

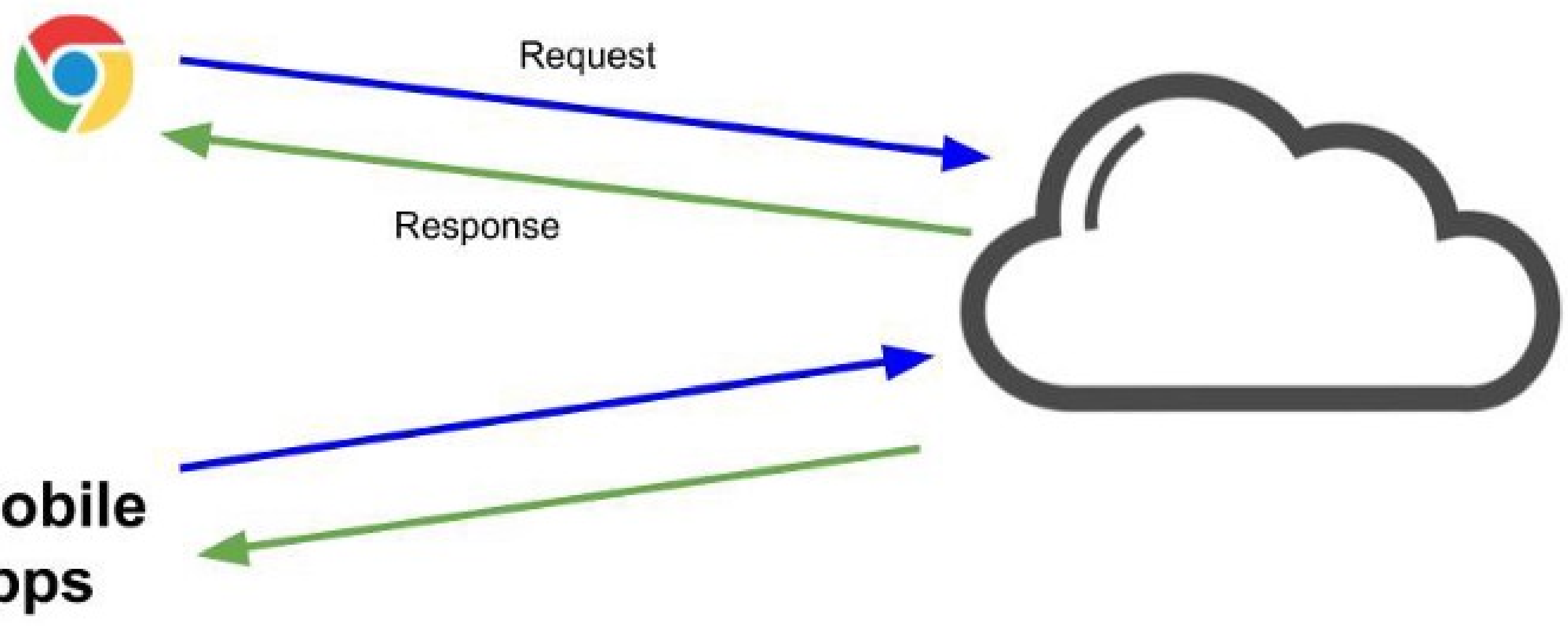
- Get social accounts as well.
  - twitter.com/YourDomainName
  - instagram.com/YourDomainName
  - ...
- Company email: @mycompany.com
  - G Suite: $6/person/month

9

After you get a domain name, make sure you also create a bunch of company accounts on social networking sites such as Twitter, Facebook, Instagram, and Snapchat.

And most companies just use G Suite for their company email, which is basically Gmail.

Online services



Internet companies build online services. Most software today is online services. If you can't access the Internet, you can't use most apps on your phone.

Online services typically follow the client/server model. The client side software sends requests to the server side and gets responses to render the UI or to perform certain tasks.

All websites are online services, obviously. We use web browsers to access websites. To some degree, mobile apps are specialized browsers.

Good Enough Servers

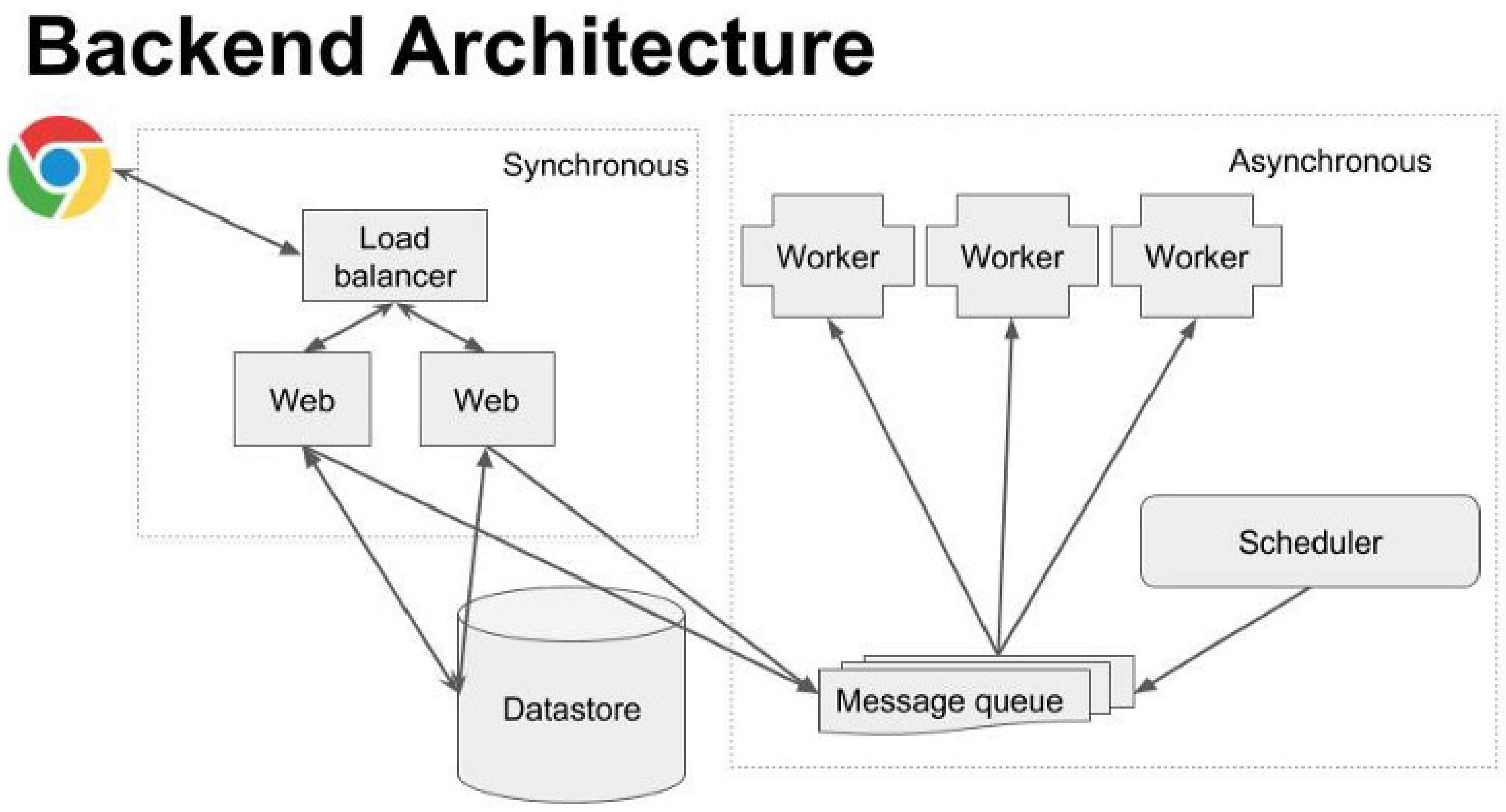
VPS (Virtual Private Servers)

- DigitalOcean or AWS lightsail
- $5/month to start

A VPS = root@ip address

On the server side, we run servers. You just need good enough servers to get started. By "servers" I mean VPS (Virtual Private Servers), which basically provides root access to an IP address. Once you ssh into an IP address (a VPS), you can do whatever you want, e.g., install software & put your code there. And you are now open for business.

For VPS, I recommend using something simple at the beginning, e.g., DigitalOcean or AWS Lightsail. At Listen Notes, we used DigitalOcean for about one year because it's cheap & easy to set up, then switched to AWS EC2 when our website got more traffic and we needed more flexibility & "production"-ish setup.



Such backend architecture is pretty common for running an online service.

The client side (e.g., browsers) sends requests. The load balancer distributes requests evenly to web servers. We typically run a lot of web servers, where the server side code is running (e.g., Rails, Django…). We need a datastore to store our data. Web servers interact with the datastore to read and write data.

On the left hand side, it's synchronous processing. Here comes a request, and the web server processes it and returns a response right away. It's synchronous.

We need asynchronous processing as well to handle non-urgent, long-running, or compute-intensive tasks. We don't want to consume compute resources for such tasks on the web server. So we typically offload such tasks for the workers to process. Web servers place messages in the message queue, and workers pick up messages to process the tasks.

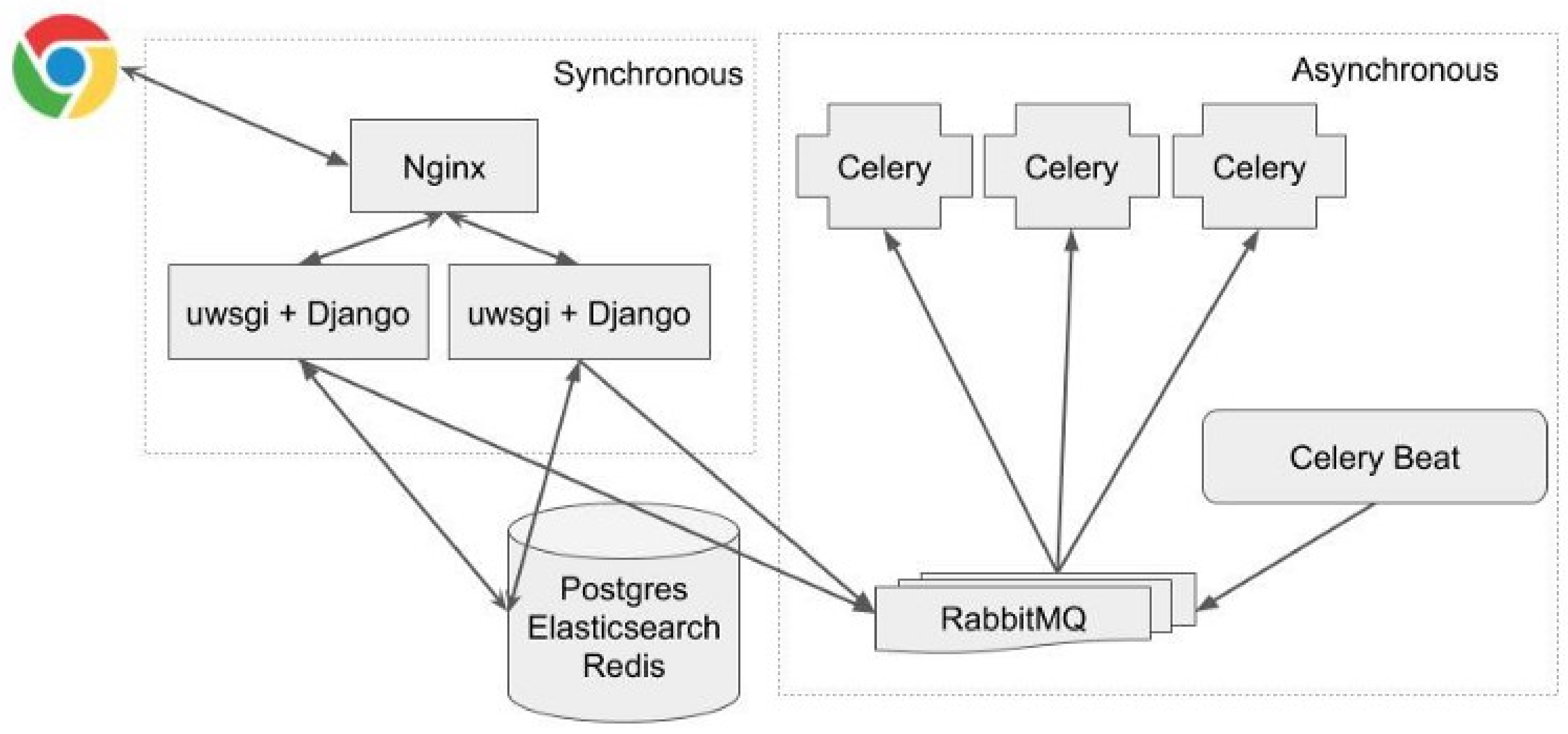
For example, we generate this kind of image for search results on Listen Notes (Example):

LISTEN NOTES
Podcasts featuring:
game of thrones
Search

This type of image-generation task is kind of compute intensive and is not urgent. So we offload it for workers to process, instead of handling them on the web servers.

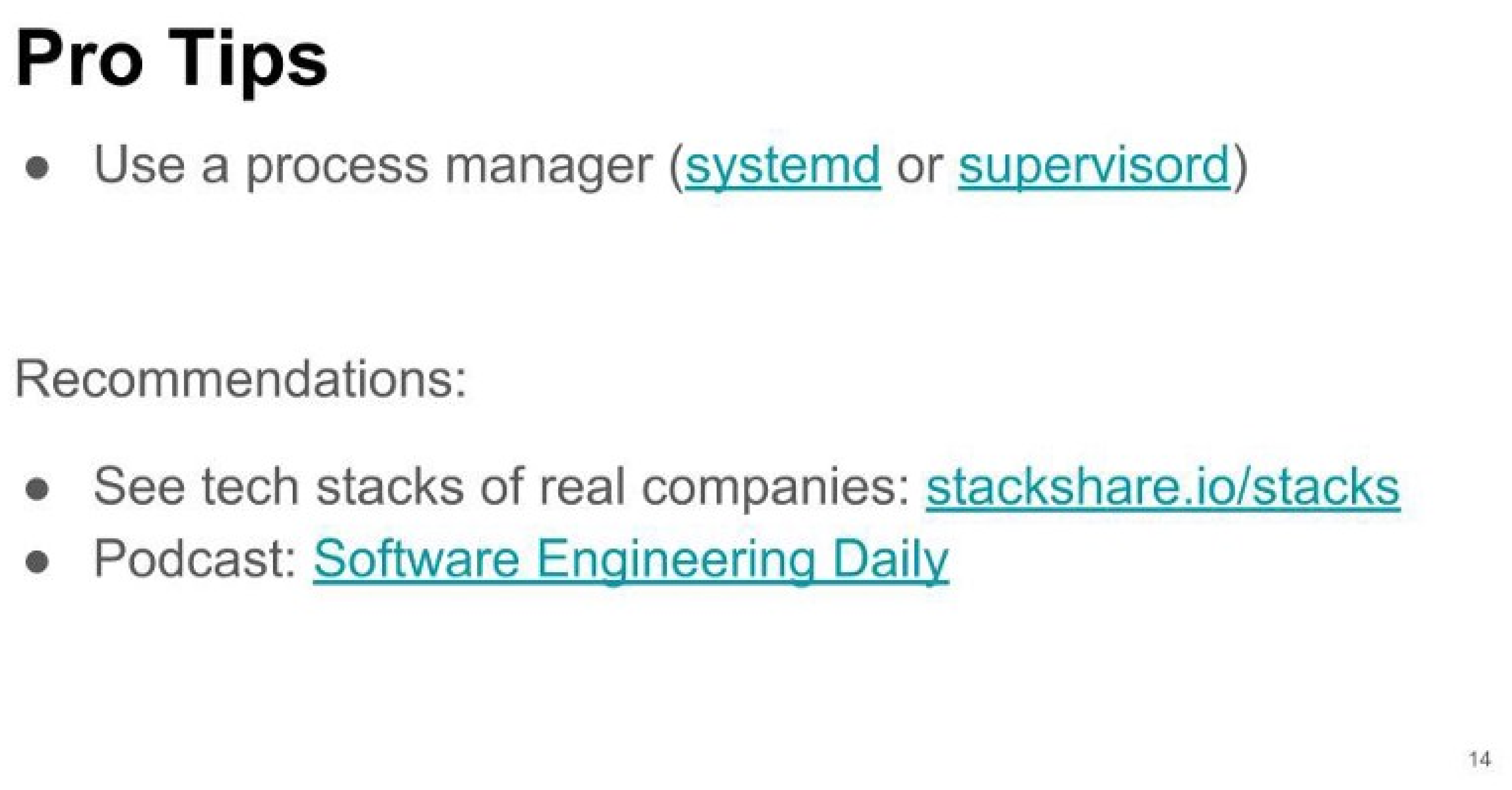
Finally, there must be a Scheduler for time-based scheduling jobs. Many companies just use cron jobs on Linux and they switch to something else when they become bigger (e.g., I built a Scheduler system for my former employer Nextdoor a few years ago to replace cron jobs). For Listen Notes, we have a lot of time-based scheduling jobs, e.g., sending Listen Alerts.

Backend Tech Stack of Listen Notes



We use Nginx as a simple load balancer. The backend code is mostly in Python/Django. We use different datastores for different purposes. We use Postgres for the main datastore, which is our single source of truth. We are a search engine, so we use Elasticsearch. We use Redis for a lot of things, but mostly for caching & implementing some "fast" features like Listen Real-Time.

On the asynchronous processing land, we use Celery, Celery Beat, and RabbitMQ.



Pro Tips

- Use a process manager (systemd or supervisord)

Recommendations:

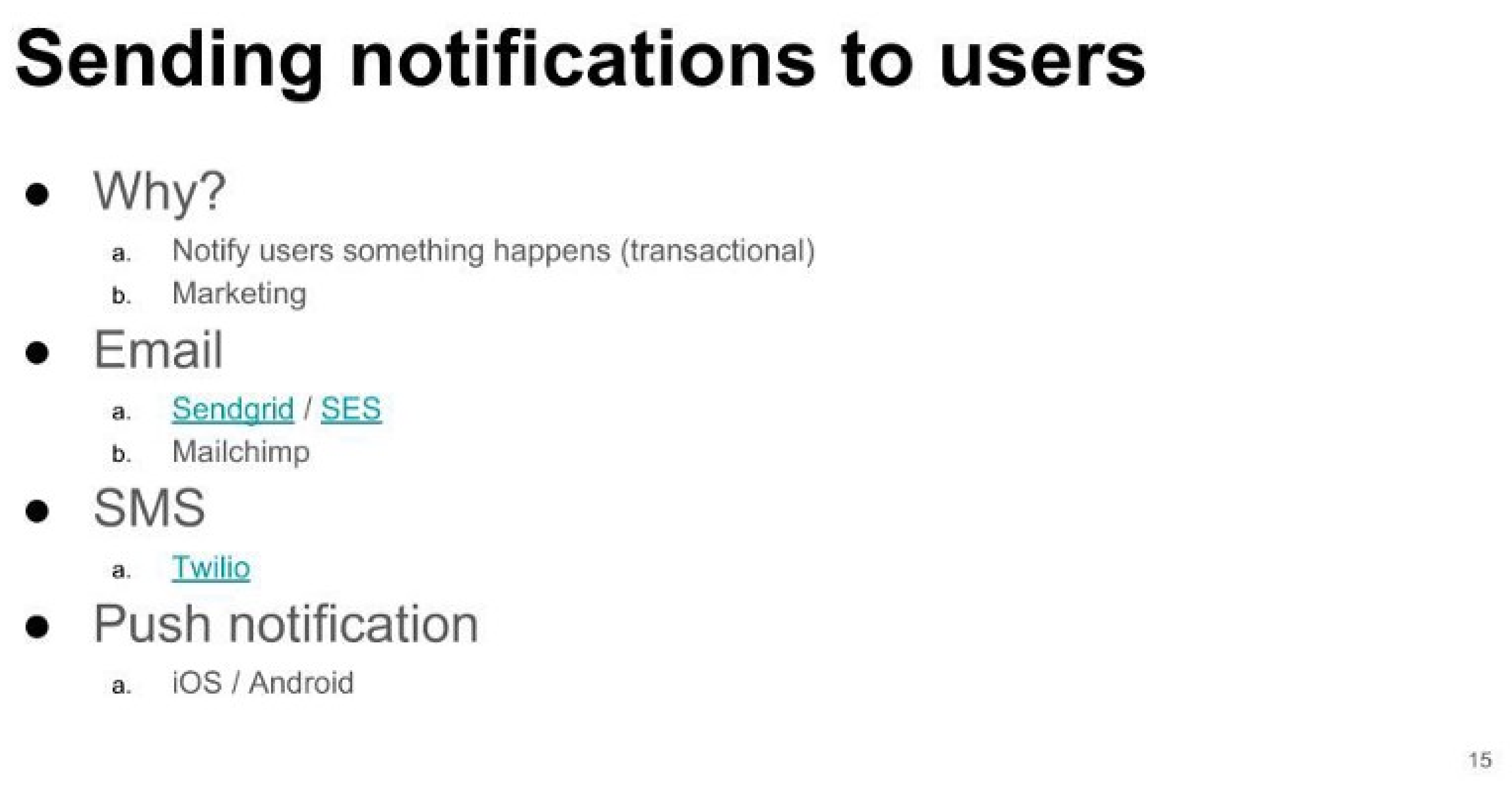
- See tech stacks of real companies: stackshare.io/stacks
- Podcast: Software Engineering Daily

14

We have to keep the server-side processes up and running 24/7. We'd better use some kind of process manager to automatically restart processes if they crash. We use Supervisor a lot at Listen Notes.

Two recommendations here:

- Learn tech stacks of real companies on stackshare.io
- Listen to Software Engineering Daily. They interview engineers from real companies, so you can learn how companies do engineering.



Sending notifications to users

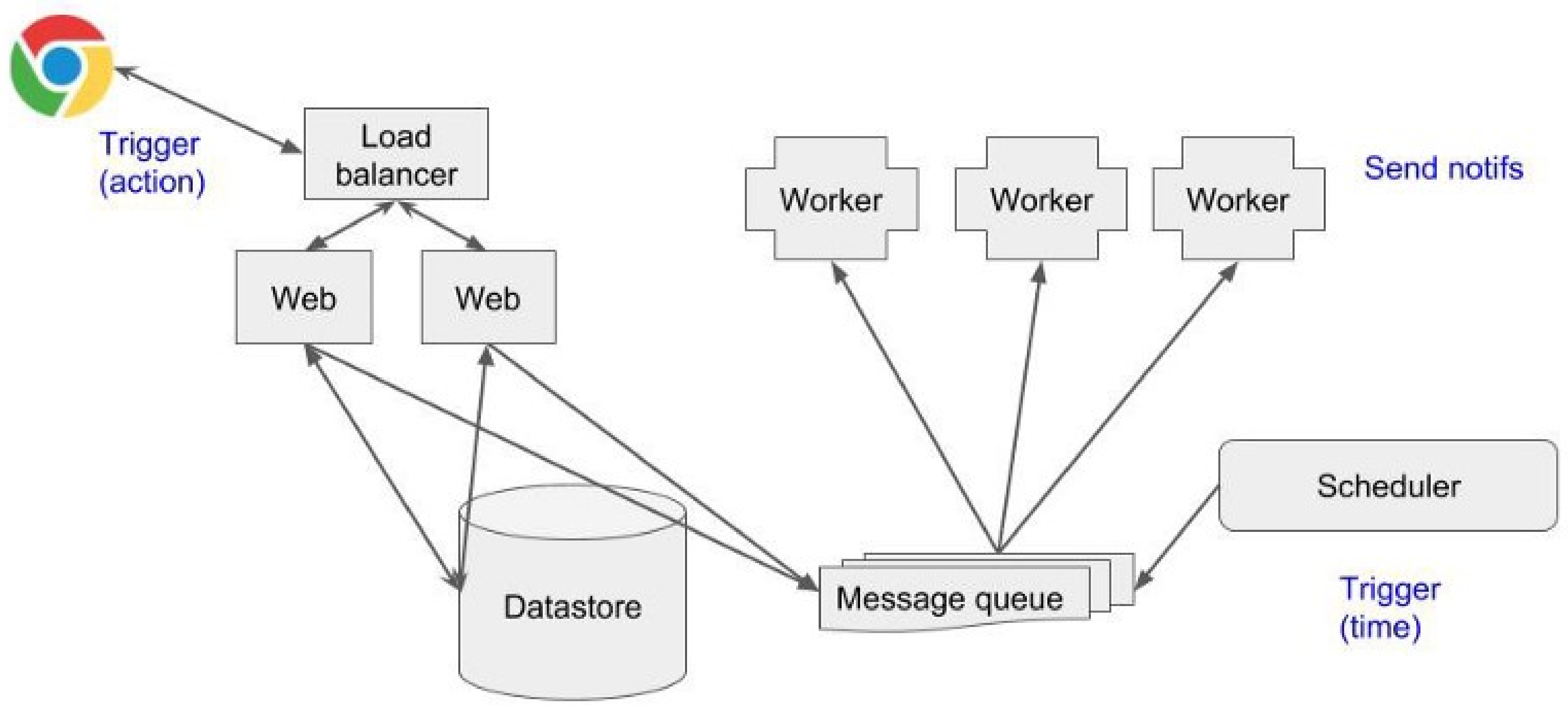
- Why?
  a. Notify users something happens (transactional)
  b. Marketing
- Email
  a. Sendgrid / SES
  b. Mailchimp
- SMS
  a. Twilio
- Push notification
  a. iOS / Android

15

As end users, we get tons of notifications from online services via email, SMS, and push notifications. When an Uber driver is approaching, we get push notifications. When we shop on Amazon, we get email notifications (typically with receipts). When our bank accounts experience problems, we get SMS notifications.

Let’s turn the table. When we build online services, how do we send notifications to users? There’s a service or API for each notification channel, e.g., SendGrid or Amazon SES for email, Twilio for SMS…

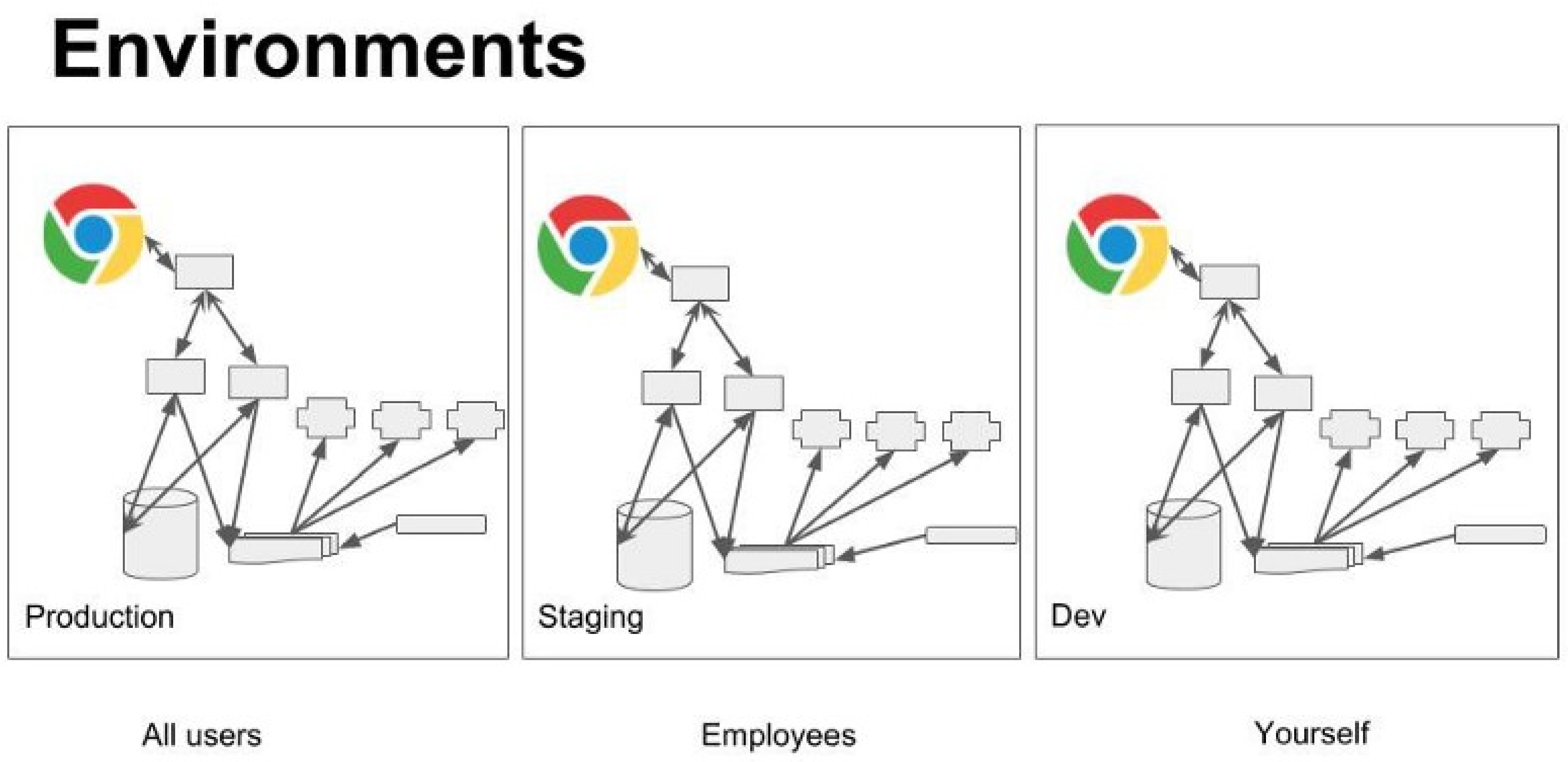
Sending notifications to users



Next, we need some kind of triggers to initiate notifications.

One type of trigger is from user actions. For example, a user can invite people to contribute to the same playlist on Listen Later. When he or she clicks the invite button, it triggers an email notification that is sent to the potential contributor. So the web server places a message in the message queue, and one of the workers picks up the message and sends the invite email later.

Another type of trigger is from time-based schedules, e.g., send emails to these 500 people at 7 a.m. every morning.



Each of the blocks in the architecture diagram represents a process or multiple processes on the operating systems. We can run those processes on one server or multiple servers.

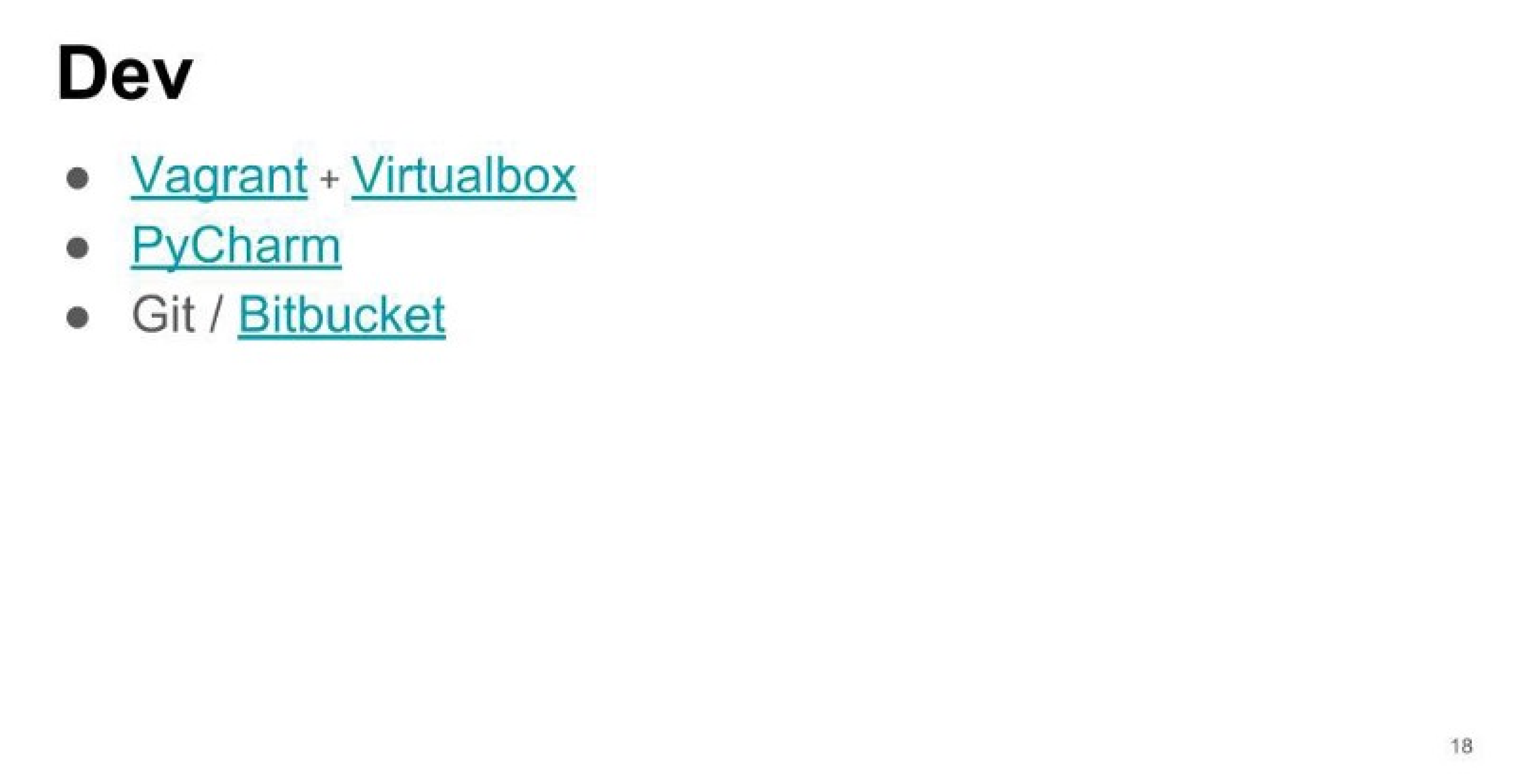
We typically create a provision of multiple sets of servers for different audiences & for different purposes.

Each set of servers runs in a separate environment.

Servers in the production environment serve live traffic. The audience is made up of real users.

Servers in the staging environment are primarily for testing. The audience is comprised of employees in the company. We need the staging environment to manually test product features before releasing the whole shebang to production.

And we need a dev environment for development purposes, which is typically used by a single developer.



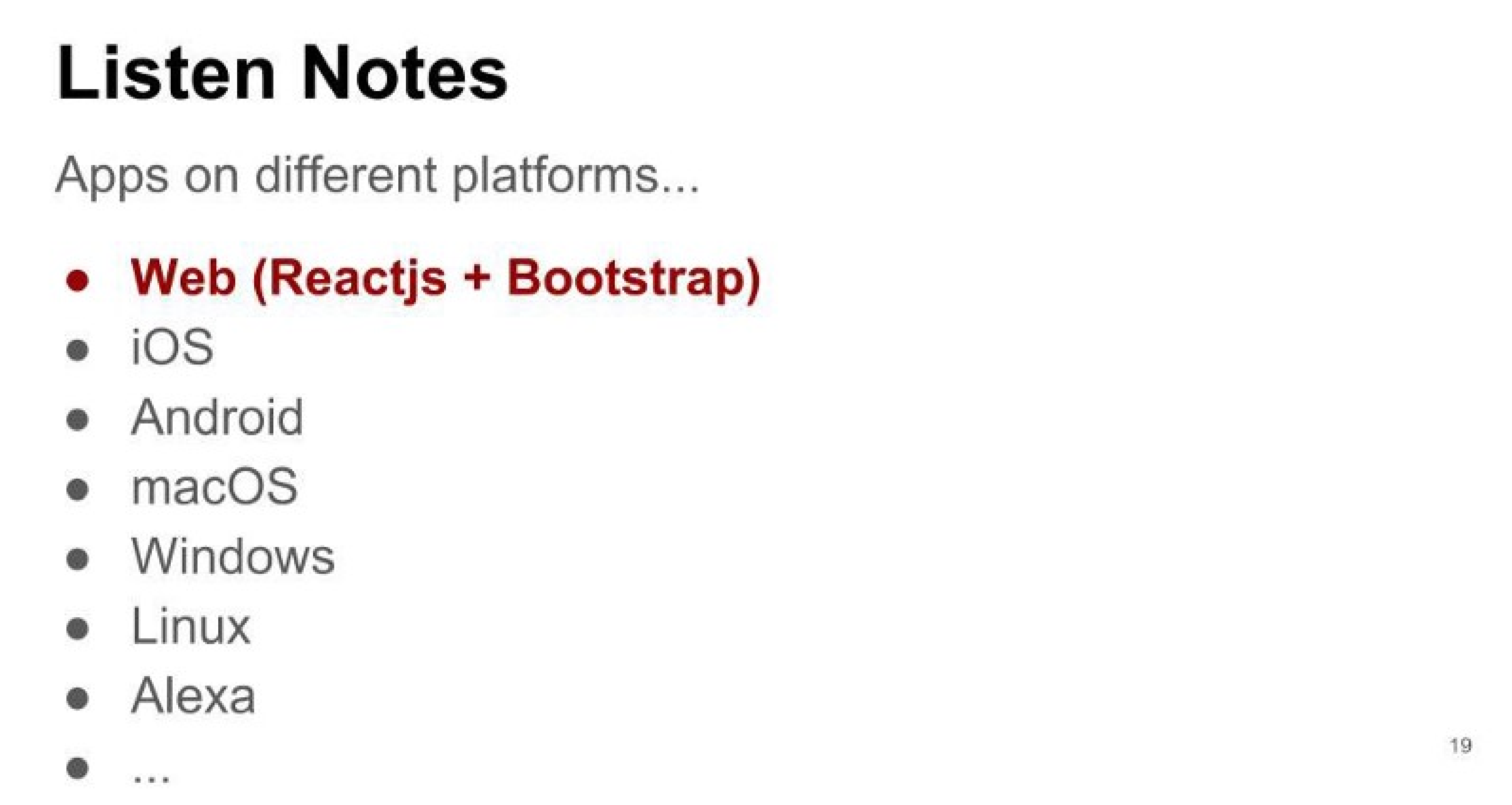
Dev

- Vagrant + Virtualbox
- PyCharm
- Git / Bitbucket

18

For Listen Notes, we use Vagrant & VirtualBox to set up a virtual machine. And we run everything inside this virtual machine.

Since the backend code of Listen Notes is primarily written in Python/Django, I use PyCharm to write code. I know, it's not VS Code or whatever cool text editors others use. But I'm 1000x more productive using PyCharm than using other text editors :) It's like some people like spicy food, while others don't. We can't blame people who don't like spicy food, right? Don't get involved in the religious war of IDEs, languages, technologies… GETTING THINGS DONE™ is more important.



Listen Notes

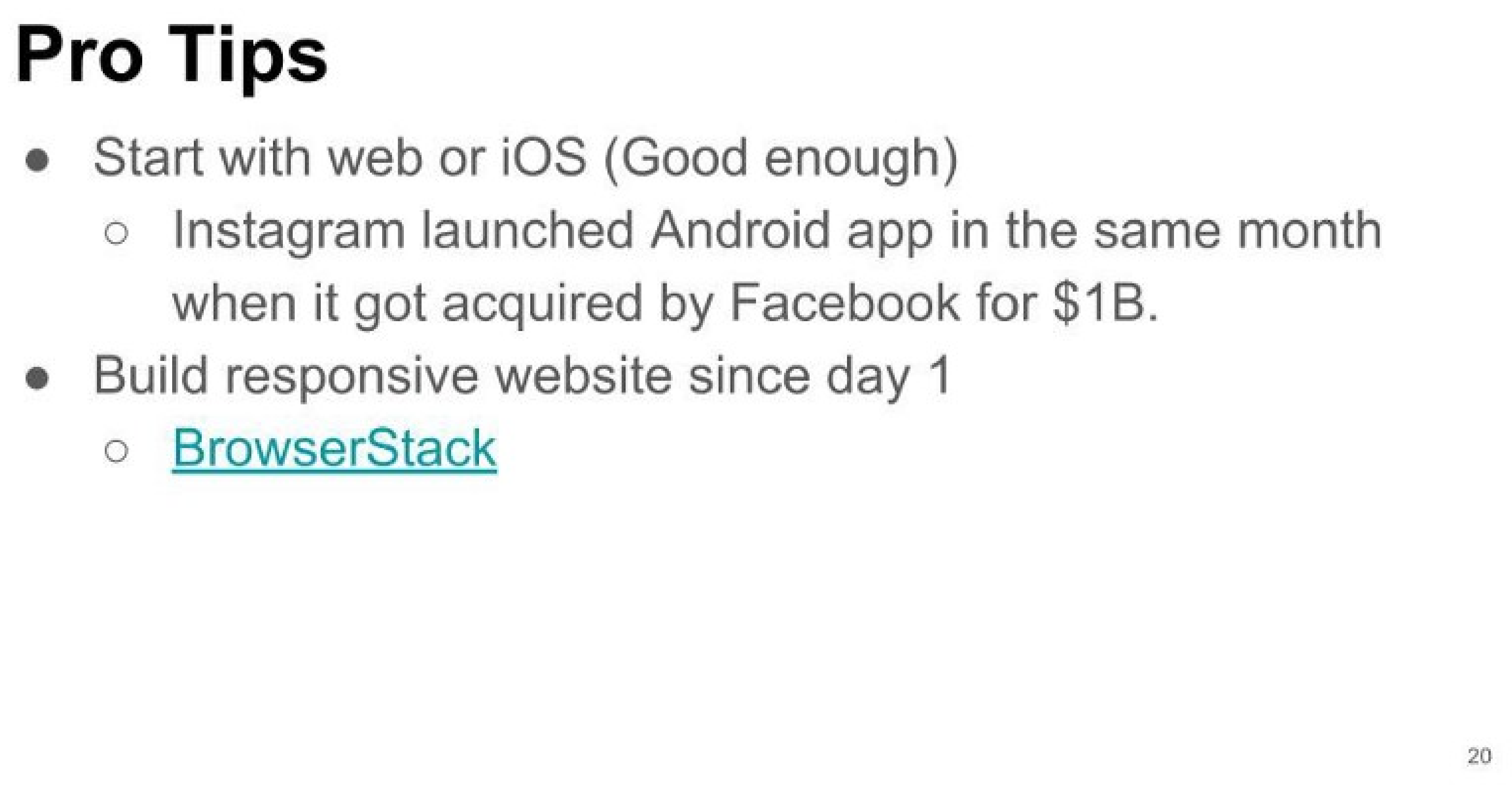
Apps on different platforms...

- **Web (Reactjs + Bootstrap)**
- iOS
- Android
- macOS
- Windows
- Linux
- Alexa
- ...

19

In terms of front-end engineering, I have very little to share here. Listen Notes has only a website. We don't have native apps (except for an experimental Just Listen app).

For the web front-end, I use the conventional Reactjs & Bootstrap. Pretty standard nowadays.



Pro Tips

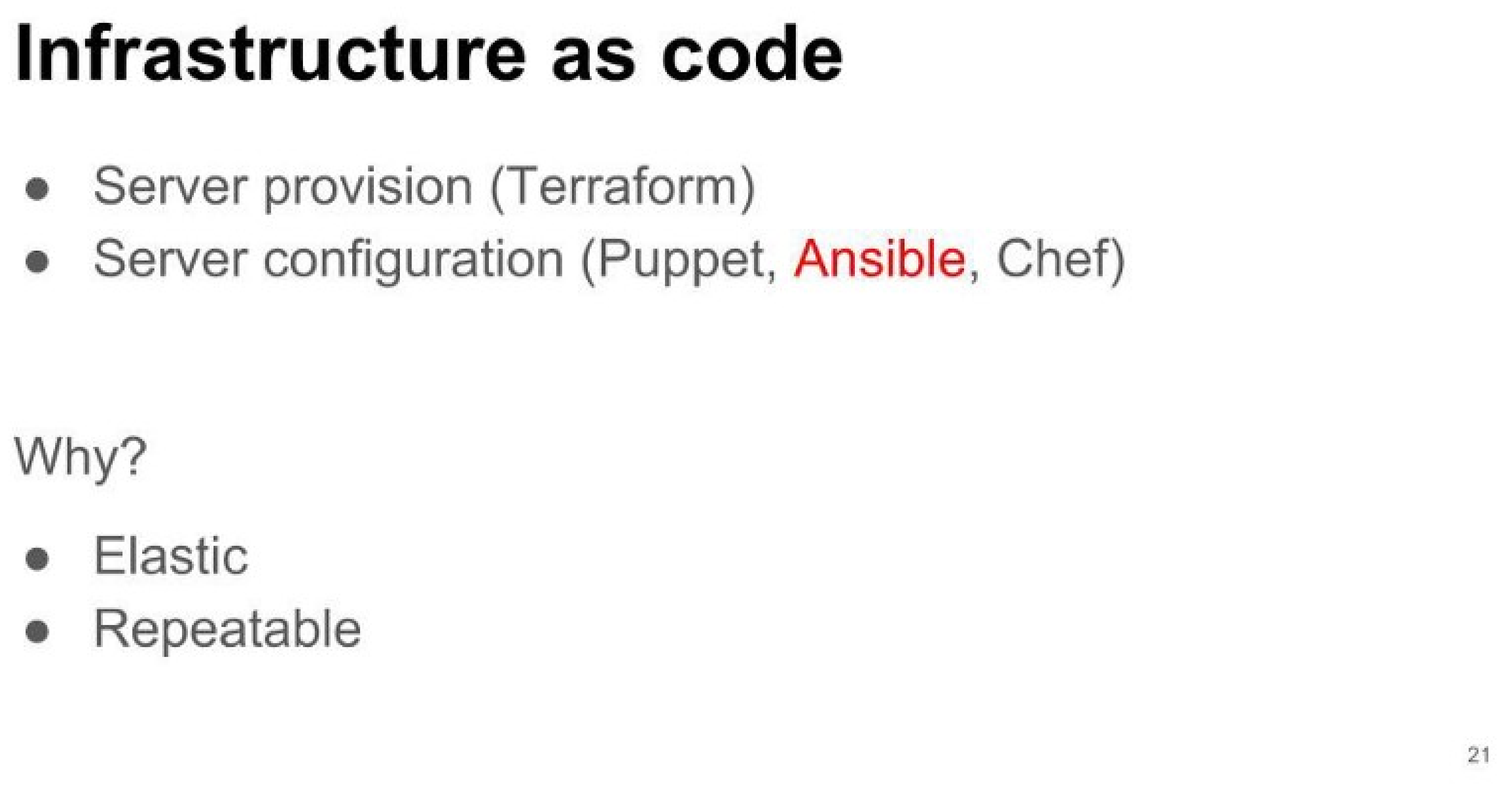
- Start with web or iOS (Good enough)
  - Instagram launched Android app in the same month when it got acquired by Facebook for $1B.
- Build responsive website since day 1
  - BrowserStack

20

If you just get started with your projects, I would suggest focusing on a single platform first. Don't go cross-platform too early. We typically have very limited resources at the very beginning. Look at Instagram: When they were an independent company, they had only an iOS app initially. And they got acquired for $1B.

If you are building a website, make sure you make it responsive from day 1. You can easily use modern browsers' developer tools to test different screen sizes (e.g., on Chrome).

You also want to test on multiple operating systems & browsers. There's a service for that: Use BrowserStack.



Infrastructure as code

- Server provision (Terraform)
- Server configuration (Puppet, Ansible, Chef)

Why?

- Elastic
- Repeatable

21

If you have more than one server, you'd better NOT install software & do configuration manually. Infrastructure as code is a common practice in internet companies nowadays. Basically, we codify the specification of servers and automate the server configuration.

For Listen Notes, we use Ansible. We need to write a bunch of yml configuration files to specify what software to install & where to put config files. When we run ansible-playbook in command line, it'll automatically configure multiple servers for us.

Nowadays, servers are elastic or even ephemeral, with on-demand configurations to suit the necessary workload. Servers come and go. You may run more servers during the daytime, when there's more traffic; and run fewer servers at night, when there's less traffic. Infrastructure as code makes this type of thing easy.

Deployment & feature release

- Weekly deployment / continuous deployment
- Feature toggle

22

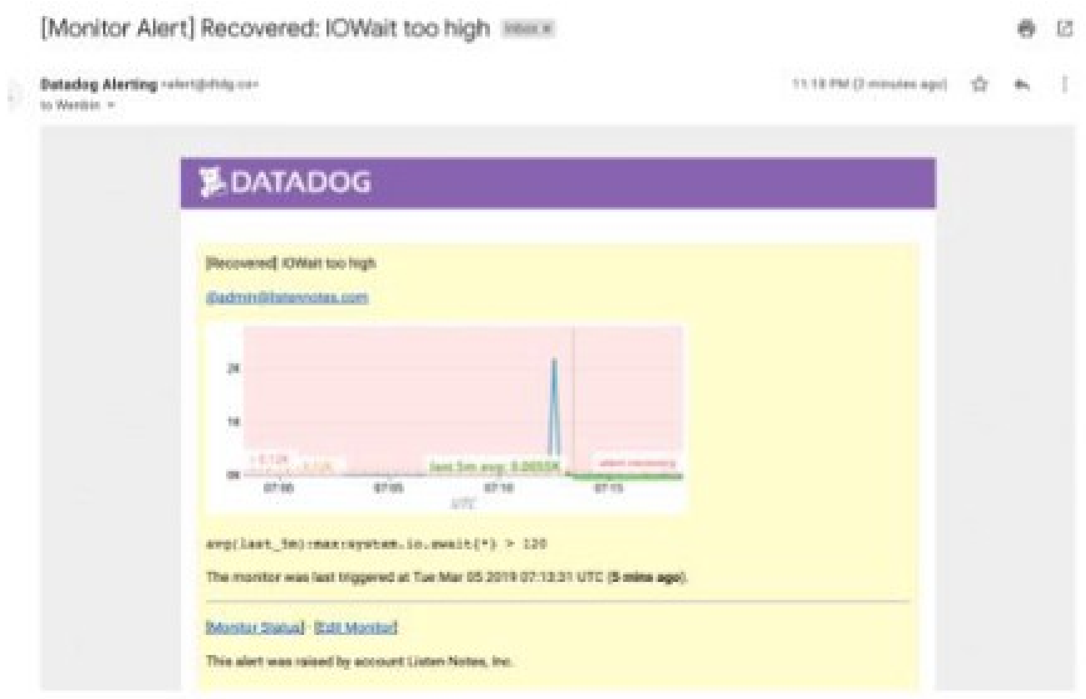
Internet companies deploy code pretty frequently nowadays, at least once per week or even many times a day. Some companies do continuous deployment, shipping code whenever there's a new git commit.

For Listen Notes, I've got a quick script to deploy code, where I can specify the deployment environment, server type, and git commit SHA as parameters. So I can push a button and deploy a specific version of code (e.g., HEAD, or any git commit) to specific servers (e.g., web, API, worker...) in a specific environment (e.g., production, staging...).

We deploy new code, but we don't necessarily release new features. Nowadays we do feature toggles in the code, which is basically some if-else statements. We can hide new code behind an if statement, and we use the feature toggle variable to control whether or not to execute the new code. We typically store the feature toggle variable in some kind of datastore, e.g., Redis. We can go very fancy here. We can turn on the new feature to 10% of users first, then 20%, then 50%, then 100%.

Monitoring & alerting

- Datadog


[Monitor Alert] Recovered: IOWait too high

Datadog Alerting <alert@dtdg.co>
to Wenbin

11:18 PM (2 minutes ago)

DATADOG

[Recovered] IOWait too high

@admin@listennotes.com

```
avg(last_5m):max:system.io.await{*} > 120
```

The monitor was last triggered at Tue Mar 05 2019 07:13:21 UTC (**5 mins ago**).

[Monitor Status] · [Edit Monitor]

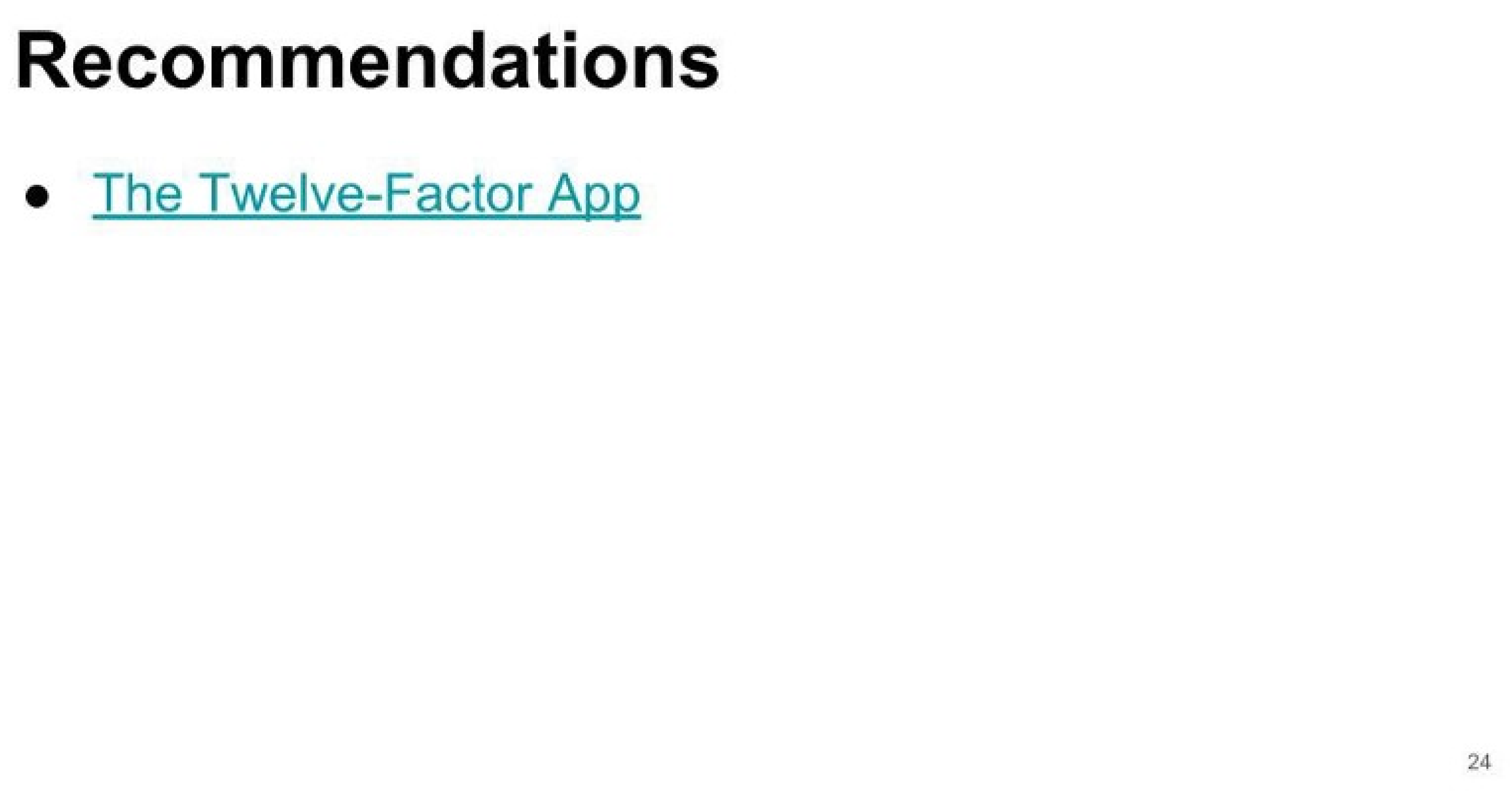
This alert was raised by account Listen Notes, Inc.

23

Online services need to be up 24/7. So it's important to have tools for monitoring & alerting.

There's a service for that!

Many companies use Datadog. We use Datadog at Listen Notes as well. We can easily build monitoring graphs to provide great visibility for servers and applications. We can also set up alerts when some metrics are abnormal (e.g., higher than a certain threshold).



Recommendations

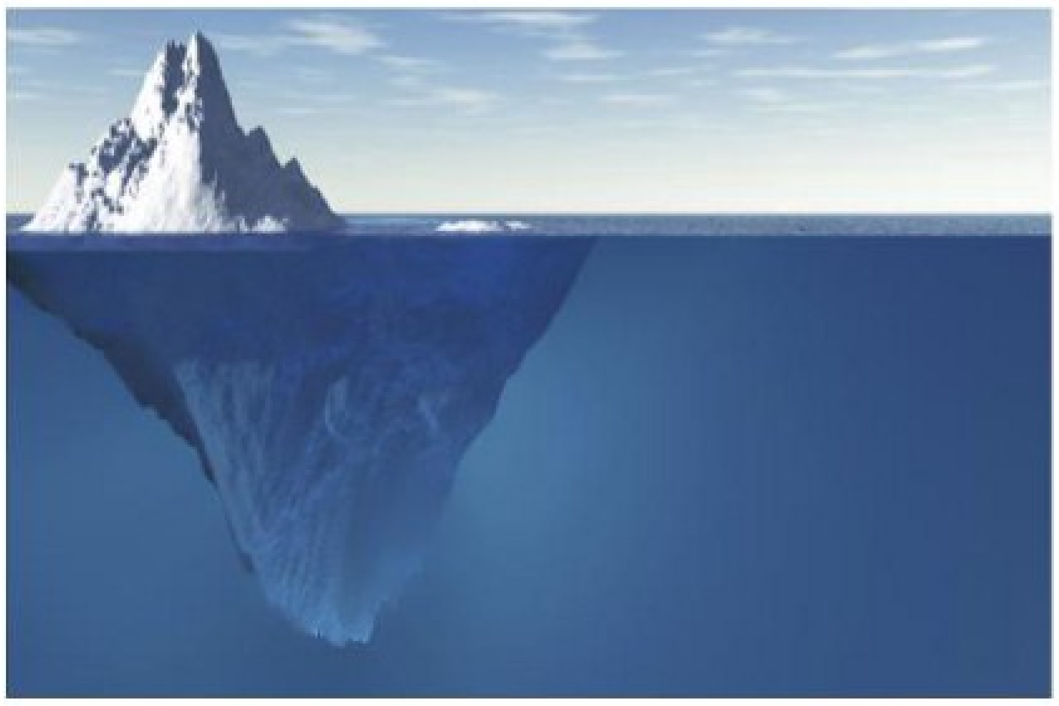
- The Twelve-Factor App

24

If you want to learn the best practices of building & operating online services, read The Twelve-Factor App.

Internal tools

- Help development
- God's view
- Fight bad actors



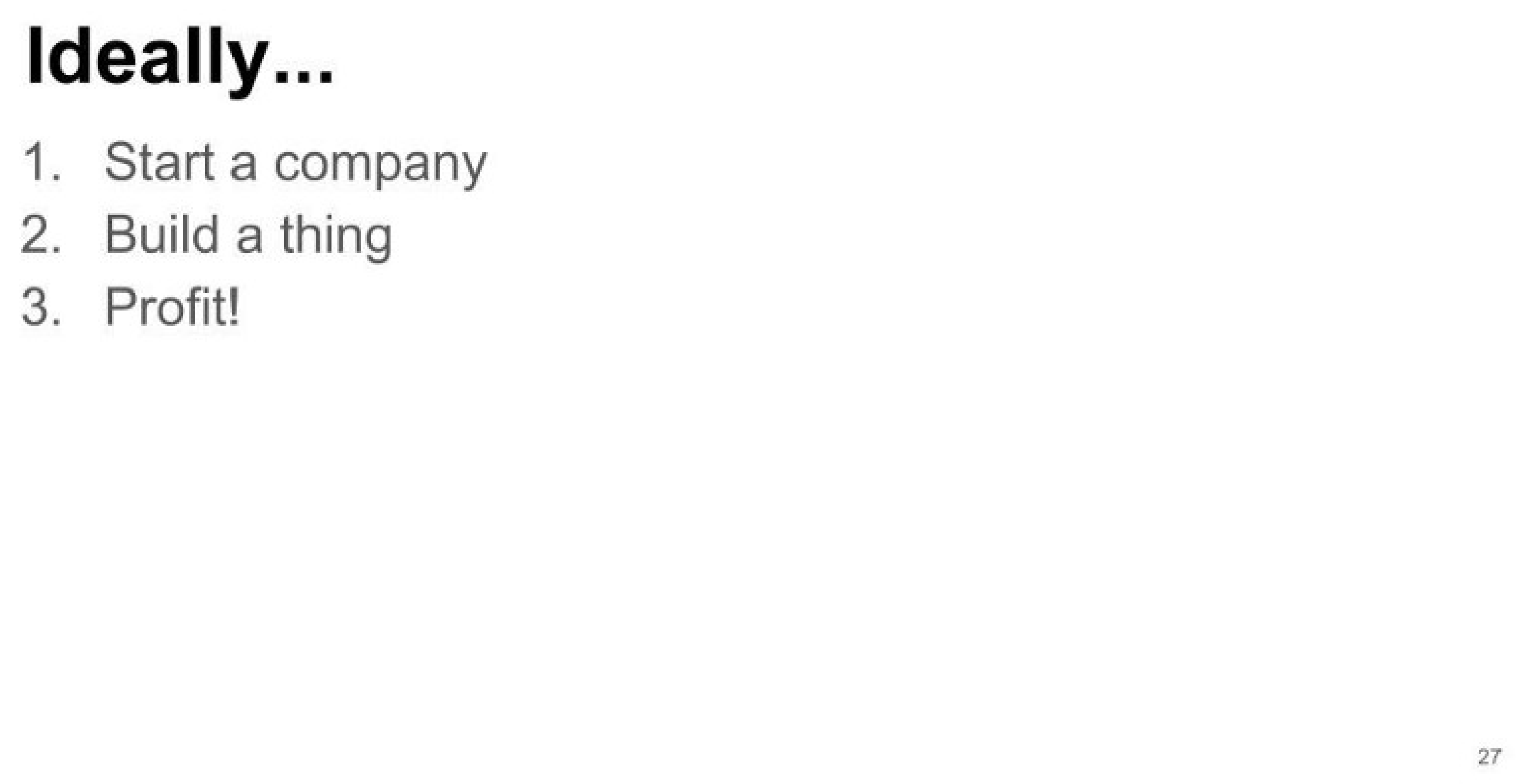
Source: TBO

25

When it comes to internal tools, I see this iceberg image. Internal tools are a big chunk of code like an iceberg lurking under the water, which is invisible to outsiders.

If you have never operated a popular online service before, you won't be aware that you need to build a lot of tools to use internally (by yourself, by your employees).

Different companies build different internal tools for various purposes. So far, I've built some internal tools to help development (e.g., preview email notification without actually sending emails), to provide God's view (e.g., see search queries, quickly pull info for a particular user...), and to fight bad actors (e.g., content moderation, detect spam...).



Ideally...

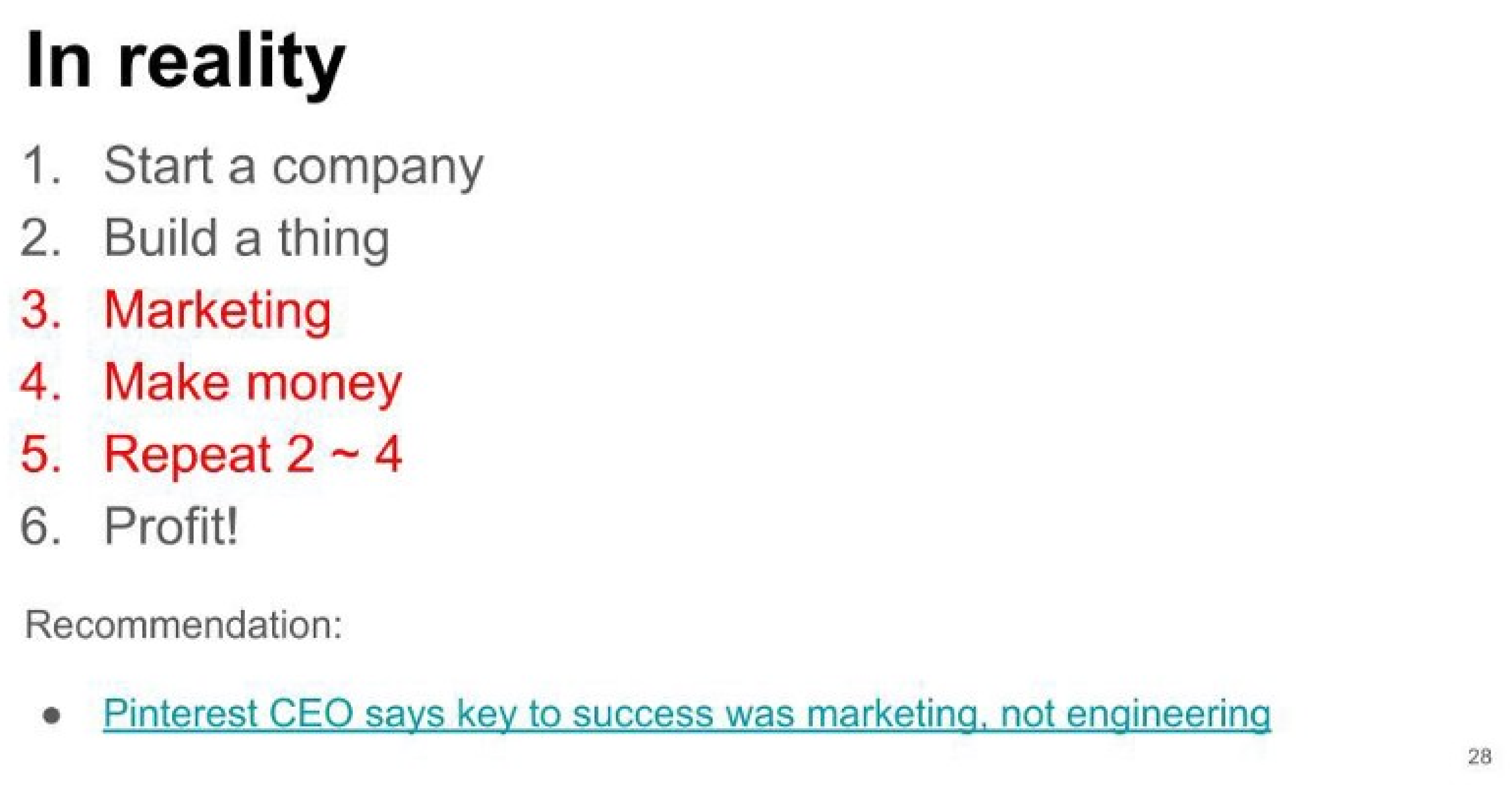
1. Start a company
2. Build a thing
3. Profit!

27

So by this point, we know how to start a company with $500 and we know how to build an online service with good enough engineering.

Profit!

Oh, wait....



In reality

1. Start a company
2. Build a thing
3. Marketing
4. Make money
5. Repeat 2 ~ 4
6. Profit!

Recommendation:

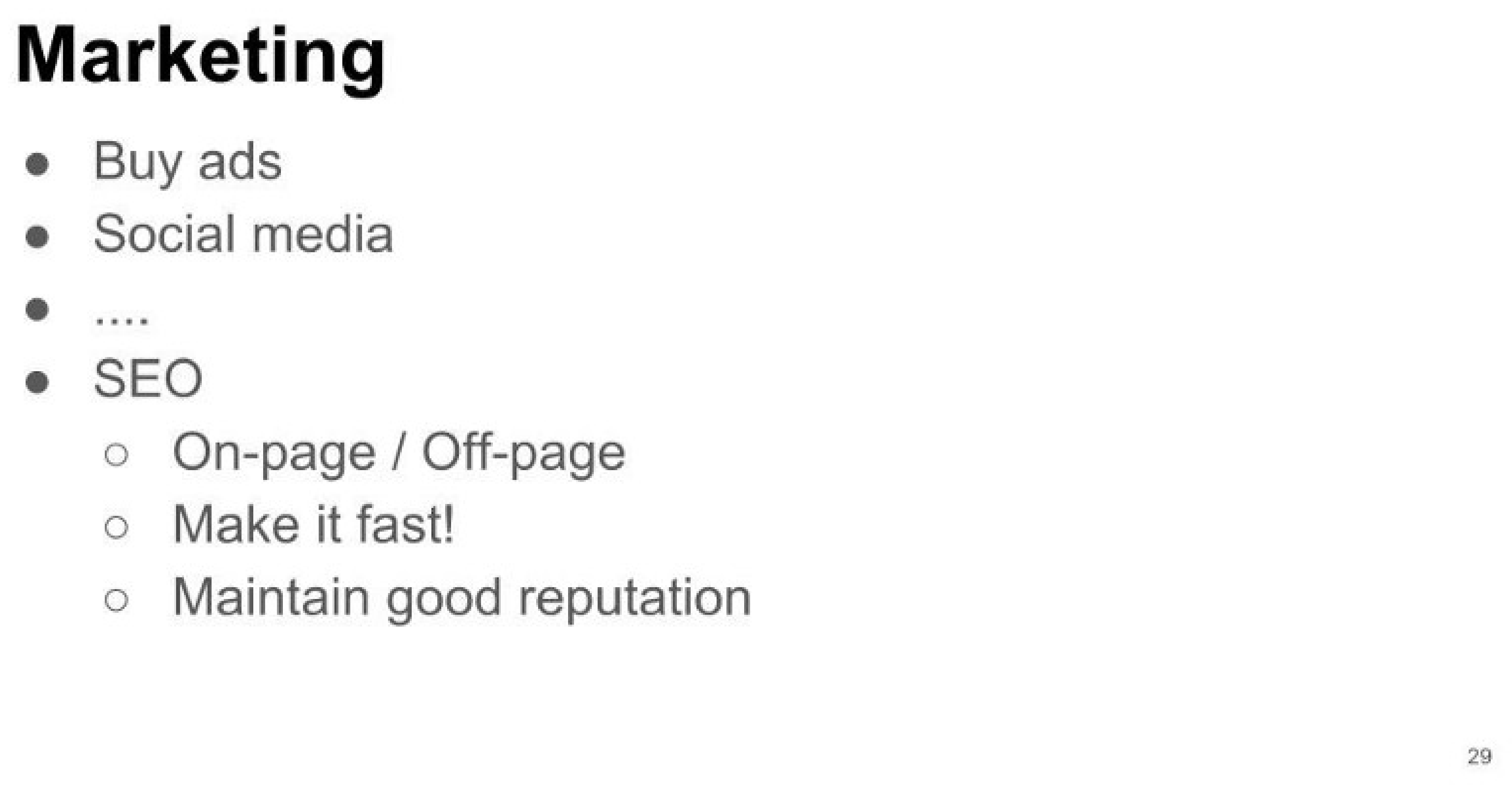
- Pinterest CEO says key to success was marketing, not engineering

28

How do people find the thing you have built? How do you make money?

Pinterest CEO says key to success was marketing, not engineering. Well, this is so true.

Engineering is deterministic. Marketing and business are non-deterministic. If you want to build an online service, you can build it. But we live in a noisy world now. Tons of things are competing for our attention. Marketing is super hard.



Marketing

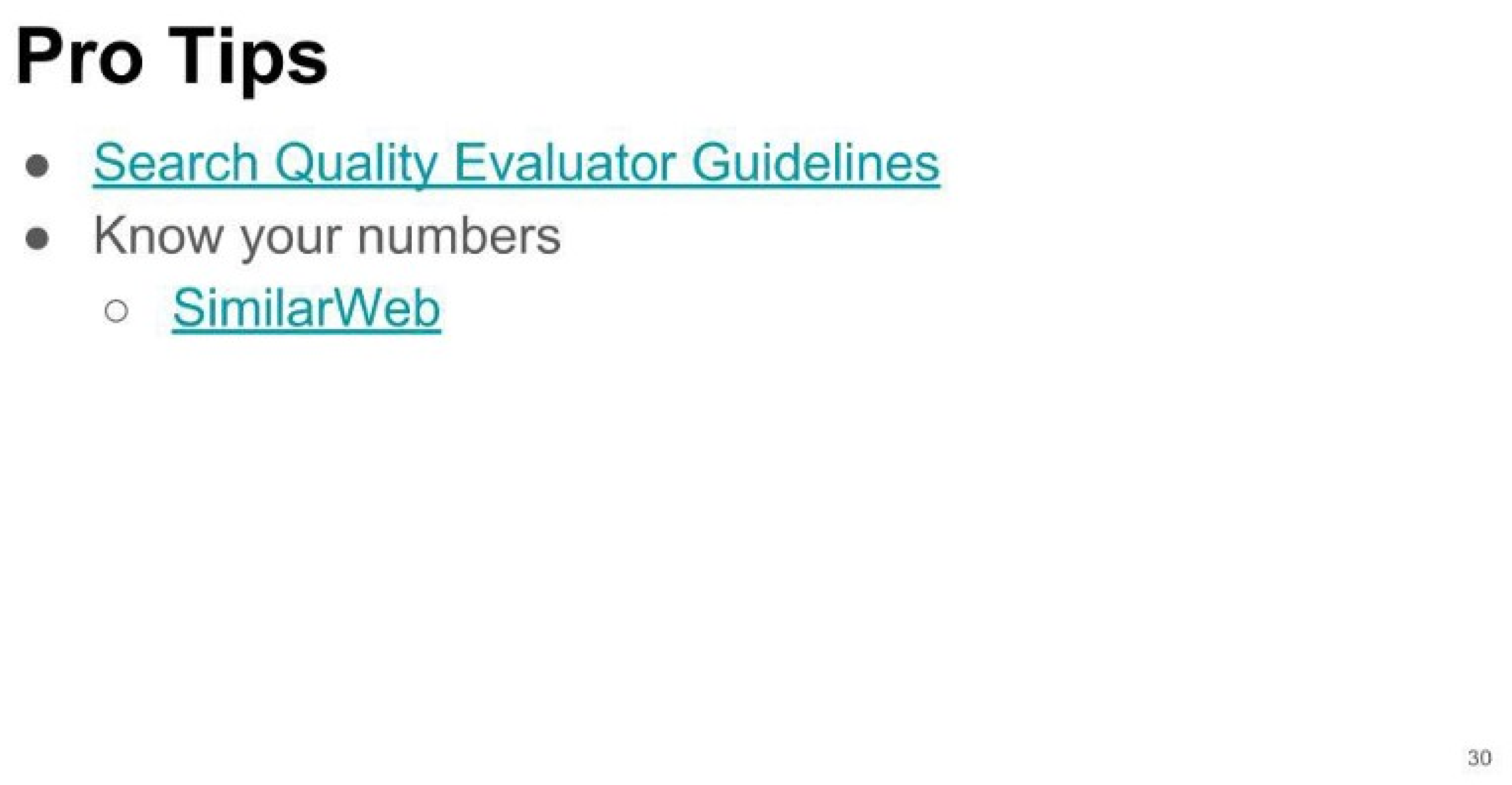
- Buy ads
- Social media
- ....
- SEO
  - On-page / Off-page
  - Make it fast!
  - Maintain good reputation

29

I'm not an expert of marketing. I'm still figuring out how to do better marketing myself...

There are multiple channels that you can use to get your messages out. Try them all. Find the most effective one. Double down on that one. A recommendation here: The 19 Channels You Can Use to Get Traction

If you want to do SEO, you can find some good tutorials on the Internet. But generally, you want to make your website as fast as possible. Google prefers fast websites nowadays.



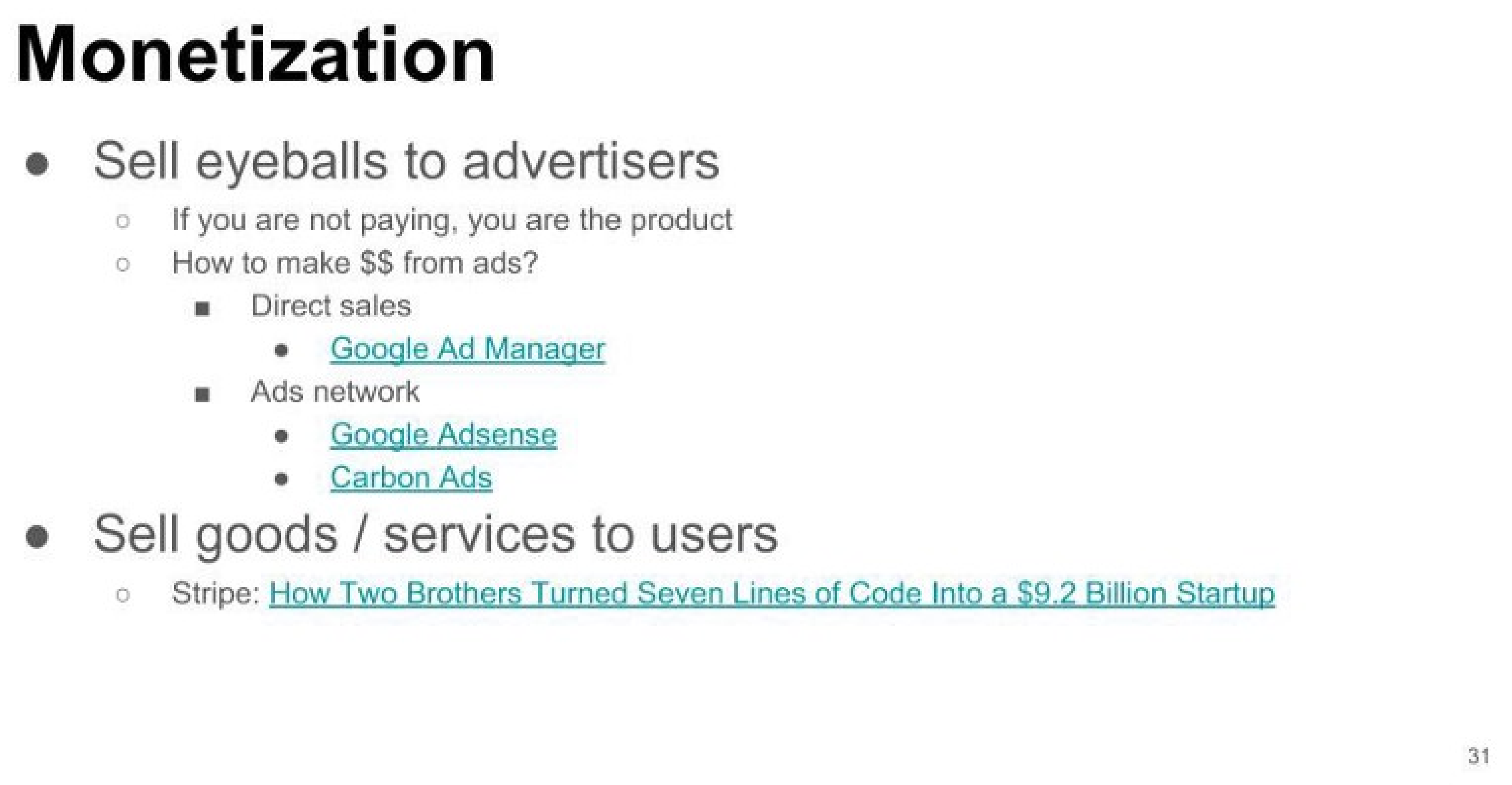
Pro Tips

- Search Quality Evaluator Guidelines
- Know your numbers
  - SimilarWeb

30

The best SEO document is from Google itself.

And if you are curious about a website's traffic, just use SimilarWeb's chrome extension. The number is not 100% accurate, but should be in the same order of magnitude :)



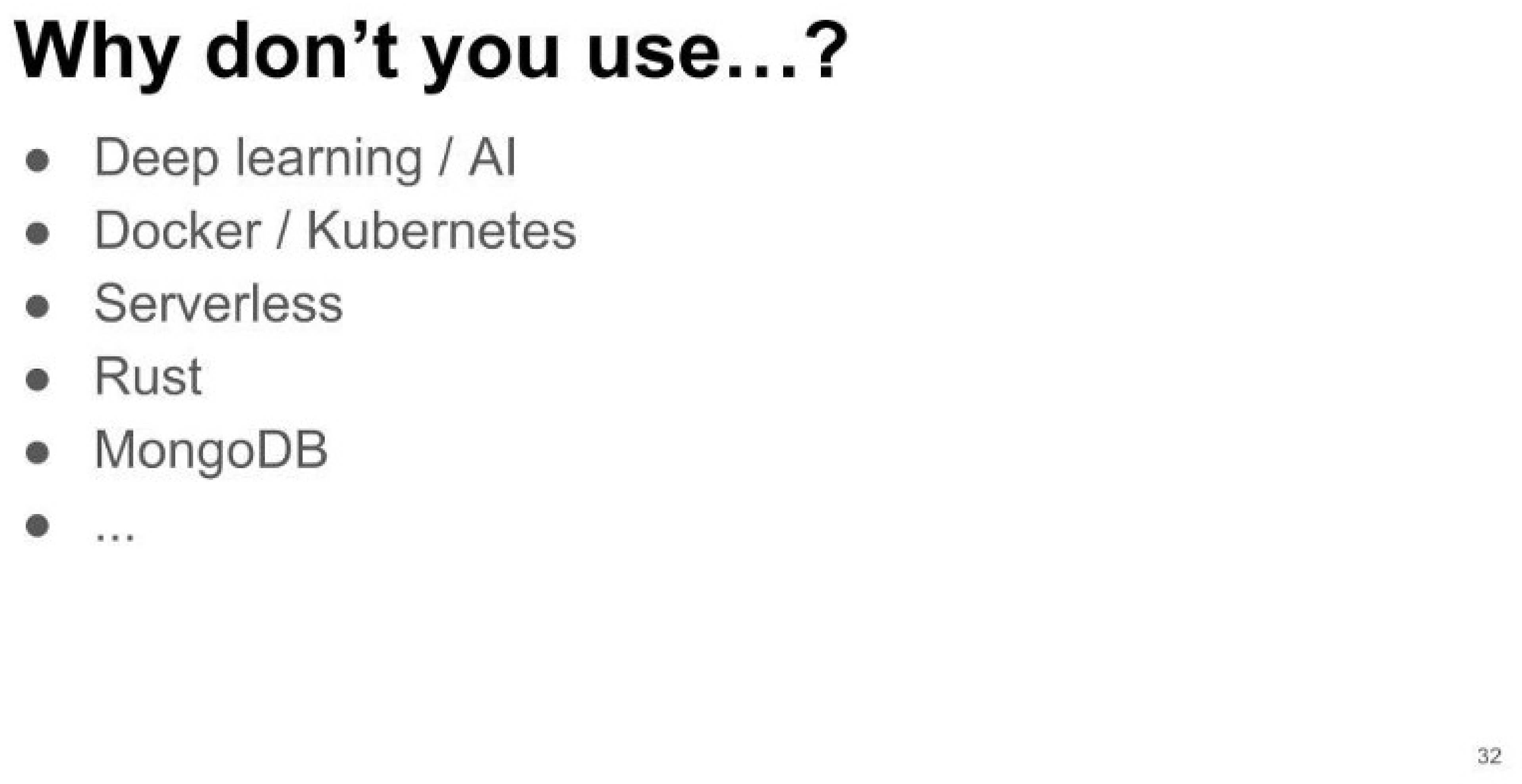
Monetization

- Sell eyeballs to advertisers
  - If you are not paying, you are the product
  - How to make $$ from ads?
    - Direct sales
      - Google Ad Manager
    - Ads network
      - Google Adsense
      - Carbon Ads
- Sell goods / services to users
  - Stripe: How Two Brothers Turned Seven Lines of Code Into a $9.2 Billion Startup

31

An Internet company makes money by selling eyeballs to advertisers or selling goods/services directly to users.

At Listen Notes, we do both. We run ads with a combination of Carbon Ads & direct sales (managed via Google Ad Manager). We also sell API to developers.

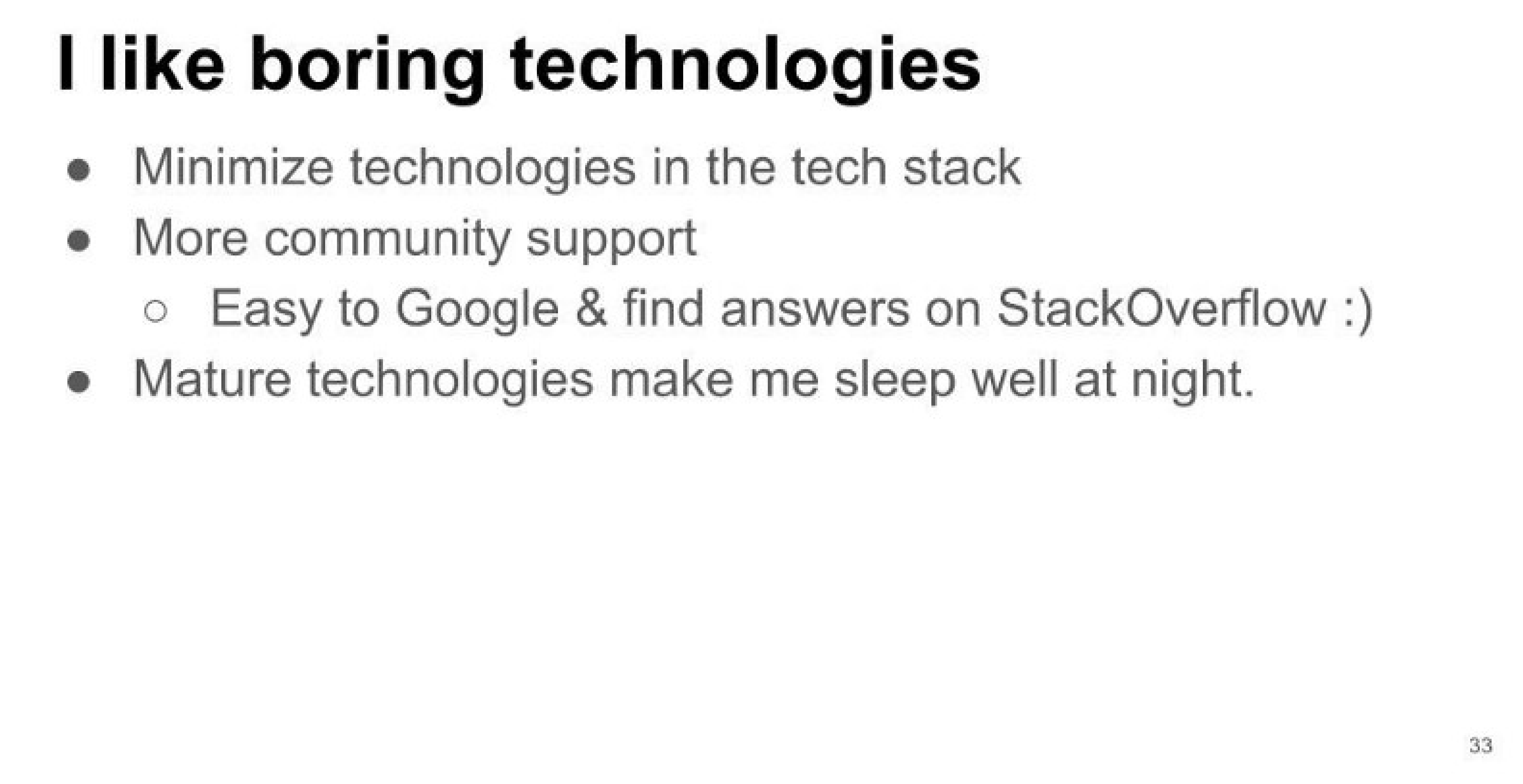


Why don't you use…?

- Deep learning / AI
- Docker / Kubernetes
- Serverless
- Rust
- MongoDB
- …

32

You may ask me why I don't use XYZ technologies. Well, I'm a practical person. The goal is to get things done, instead of doing tech for the sake of doing tech.



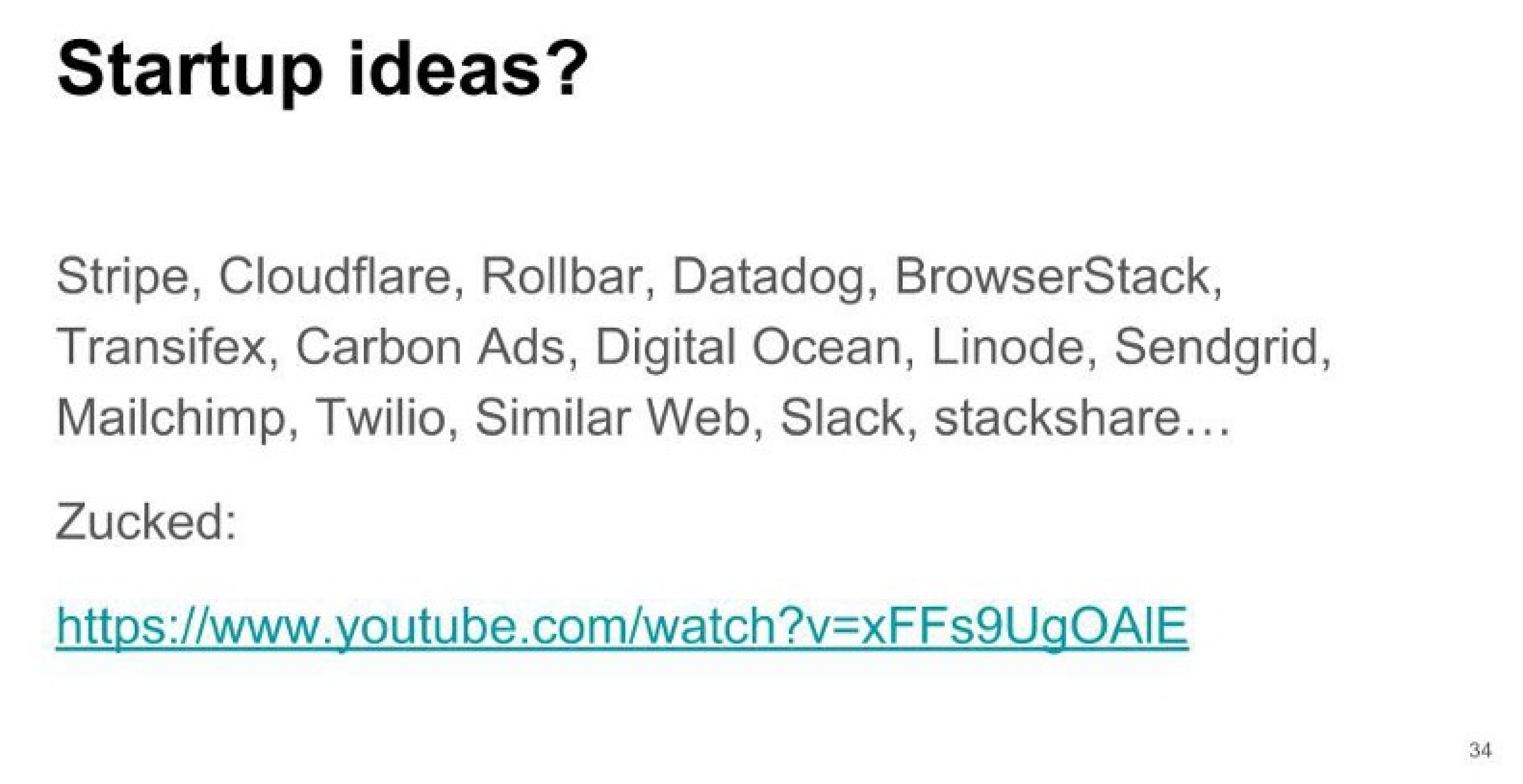
I like boring technologies

- Minimize technologies in the tech stack
- More community support
  - Easy to Google & find answers on StackOverflow :)
- Mature technologies make me sleep well at night.

33

In particular, I prefer boring technologies, which typically have existed for many years or even decades.

Software engineering nowadays is mostly Google and StackOverflow-driven :) If you need help, you can find more information for old & mature technologies from Google & StackOverflow — but like many things in our lives, this is not always true.



Startup ideas?

Stripe, Cloudflare, Rollbar, Datadog, BrowserStack, Transifex, Carbon Ads, Digital Ocean, Linode, Sendgrid, Mailchimp, Twilio, Similar Web, Slack, stackshare…

Zucked:

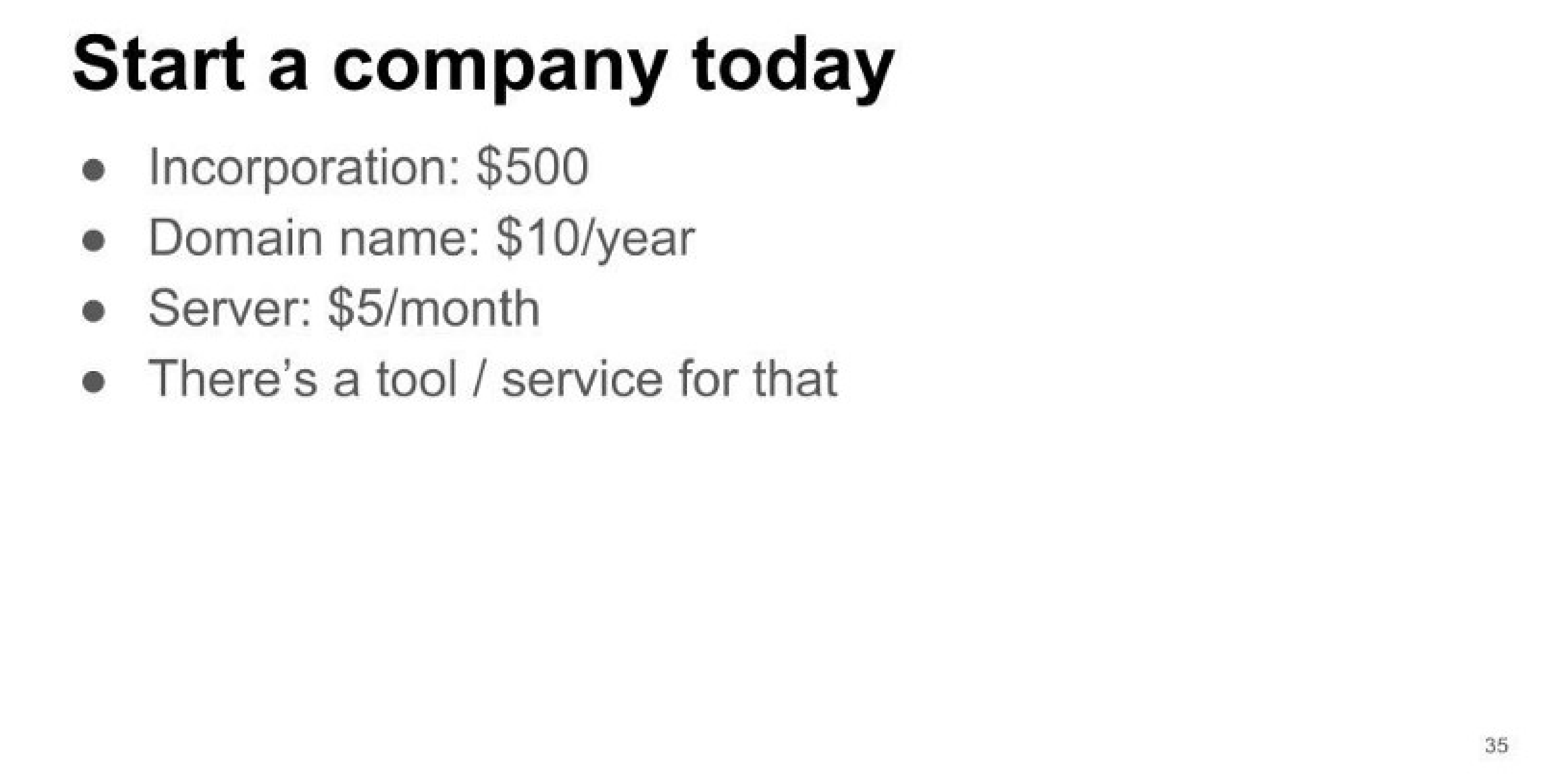
https://www.youtube.com/watch?v=xFFs9UgOAlE

34

I need to tell you bad news: It's impossible for you to come up with a 100% original startup ideas nowadays. If you think your idea is unique and original, then it's more likely that you don't read enough books or don't listen to enough podcasts :)

When I mentioned "there's a tool/service for that," it's a startup itself. You can borrow their ideas, and build a better version. Or you can tackle similar problems from a different angle.

Oh, and here's a video about Facebook's tech in 2005: https://www.youtube.com/watch?v=xFFs9UgOAlE



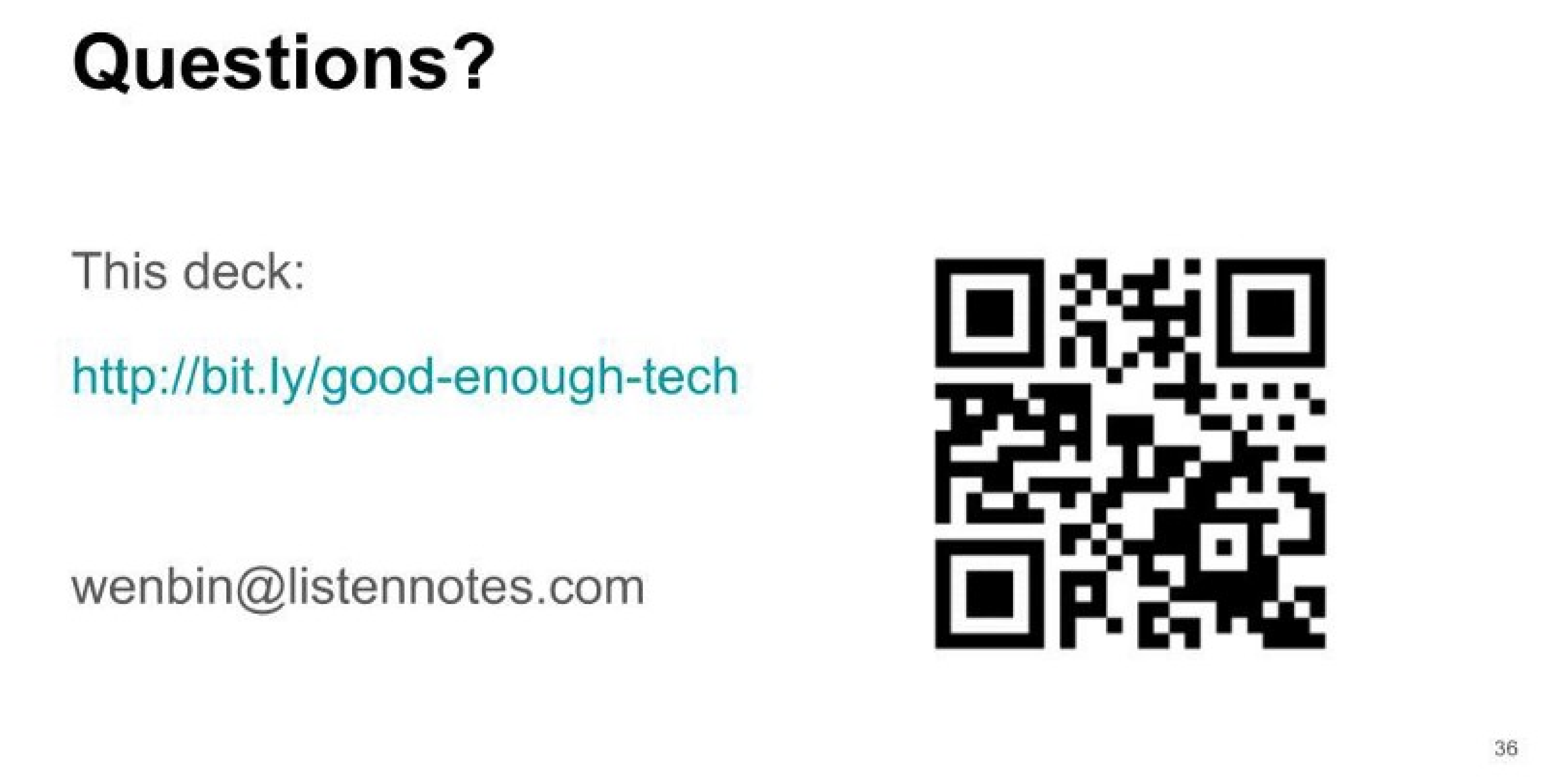
Start a company today

- Incorporation: $500
- Domain name: $10/year
- Server: $5/month
- There’s a tool / service for that

35

Okay. You can start a company TODAY!

Eighty Percent of Success Is Showing Up



Questions?

This deck:

http://bit.ly/good-enough-tech

wenbin@listennotes.com

36

You can find the deck here: http://bit.ly/good-enough-tech

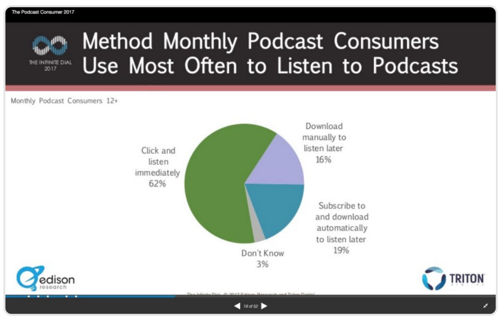
And you can always talk to me asynchronously via email :)

PREVIOUS ARTICLE:
[Listen Explorer: Get similar podcasts]()

NEXT ARTICLE:
[Listen API v2: simple pricing & same endpoints]()

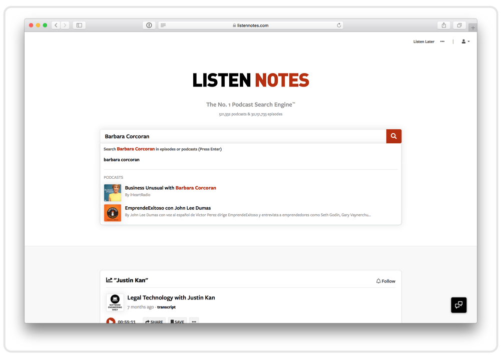
FEATURED BLOG POSTS »

The boring technology behind a one-person Internet company

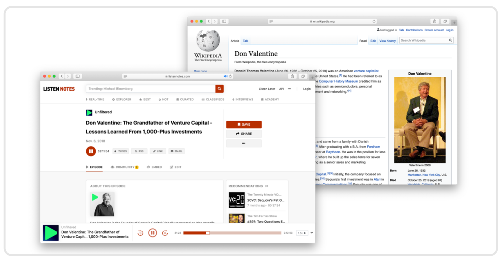

The Podcast Consumer 2017

Method Monthly Podcast Consumers Use Most Often to Listen to Podcasts

How do people use Listen Notes?



Why a standalone podcast search engine website?



Why Podcasts Are My New Wikipedia—the Perfect Informal Learning Resource

PLAYLISTS FROM COMMUNITY »

Cultura, società e politica

1 episodes and 78 podcasts

Updated 3 hours ago

Jeanne's audio diary playlist

4 episodes and 2 podcasts

Updated 5 hours ago

Jeanne's Covid 19 playlist

9 episodes and 4 podcasts

Updated 5 hours ago

Listen Notes for Chrome

The fastest way to find podcasts!