💟 Hello, brave archive readers! 💟

- This talk was given at purplecon 2024
  - Find it in video form here https://www.youtube.com/watch?v=IsHNK0m7D2U
  - Find more about purplecon here https://purplecon.org

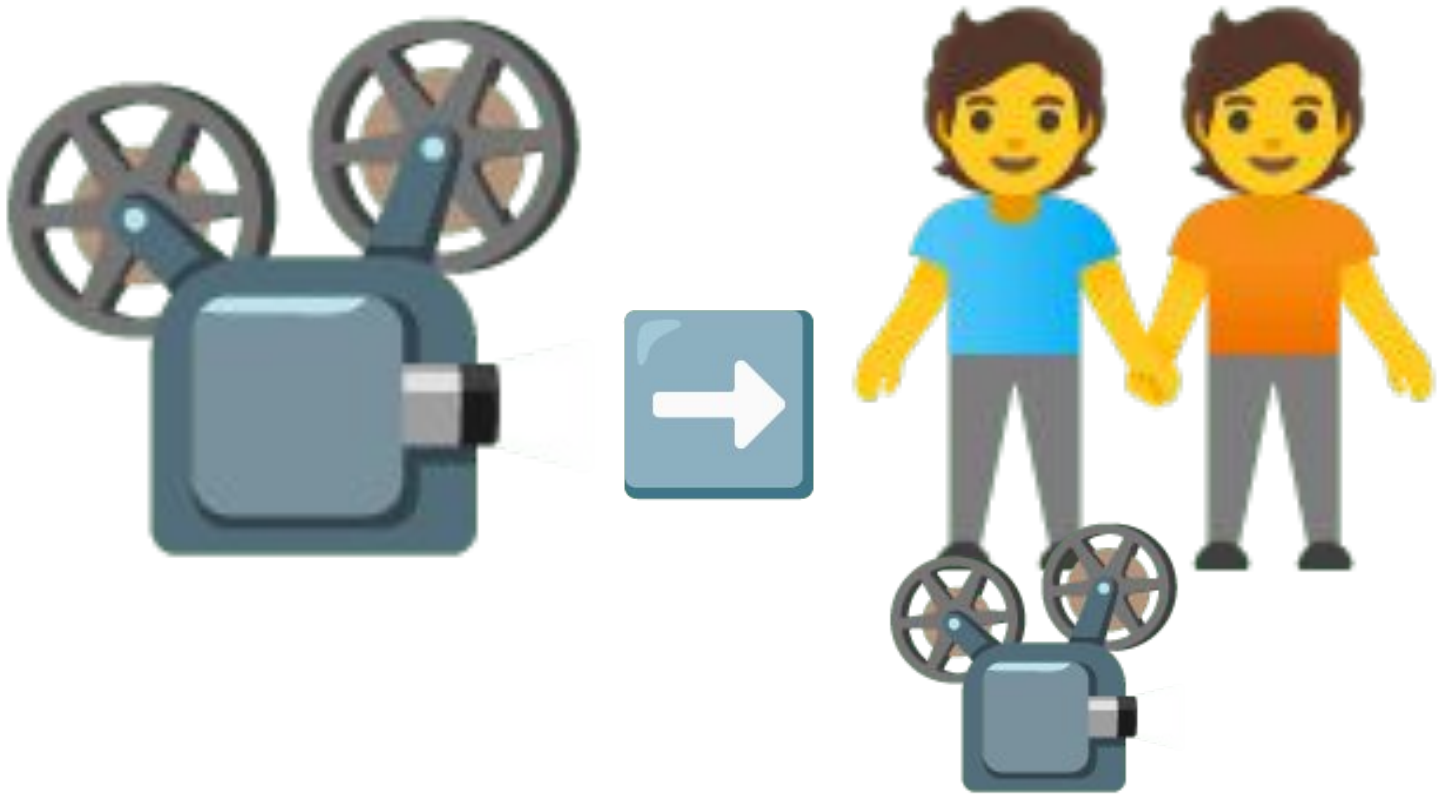
- What should you take away from the talk?
  - Give yourself permission to question the "well-lit path" that big cloud pushes on you
  - Simple is nearly always better, and you can design software in lots of ways

- Can you ask me questions or have me consult for you(r company)?
  - Yes! Hit me up at @samthor on a bunch of things
  - Or find me online at https://samthor.au

@samthor at purplecon 2024

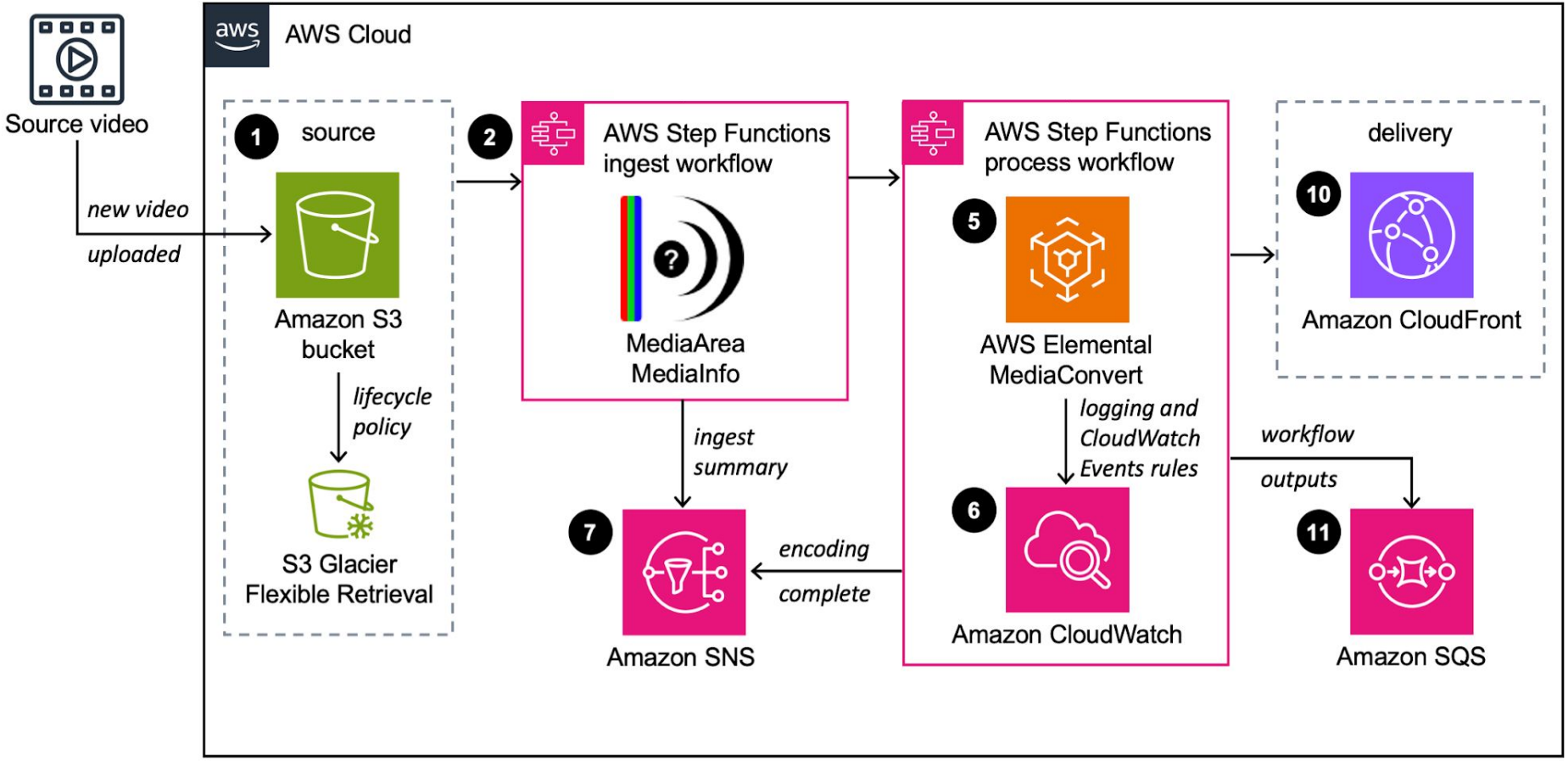
the dangers of serverless & IAM proliferation

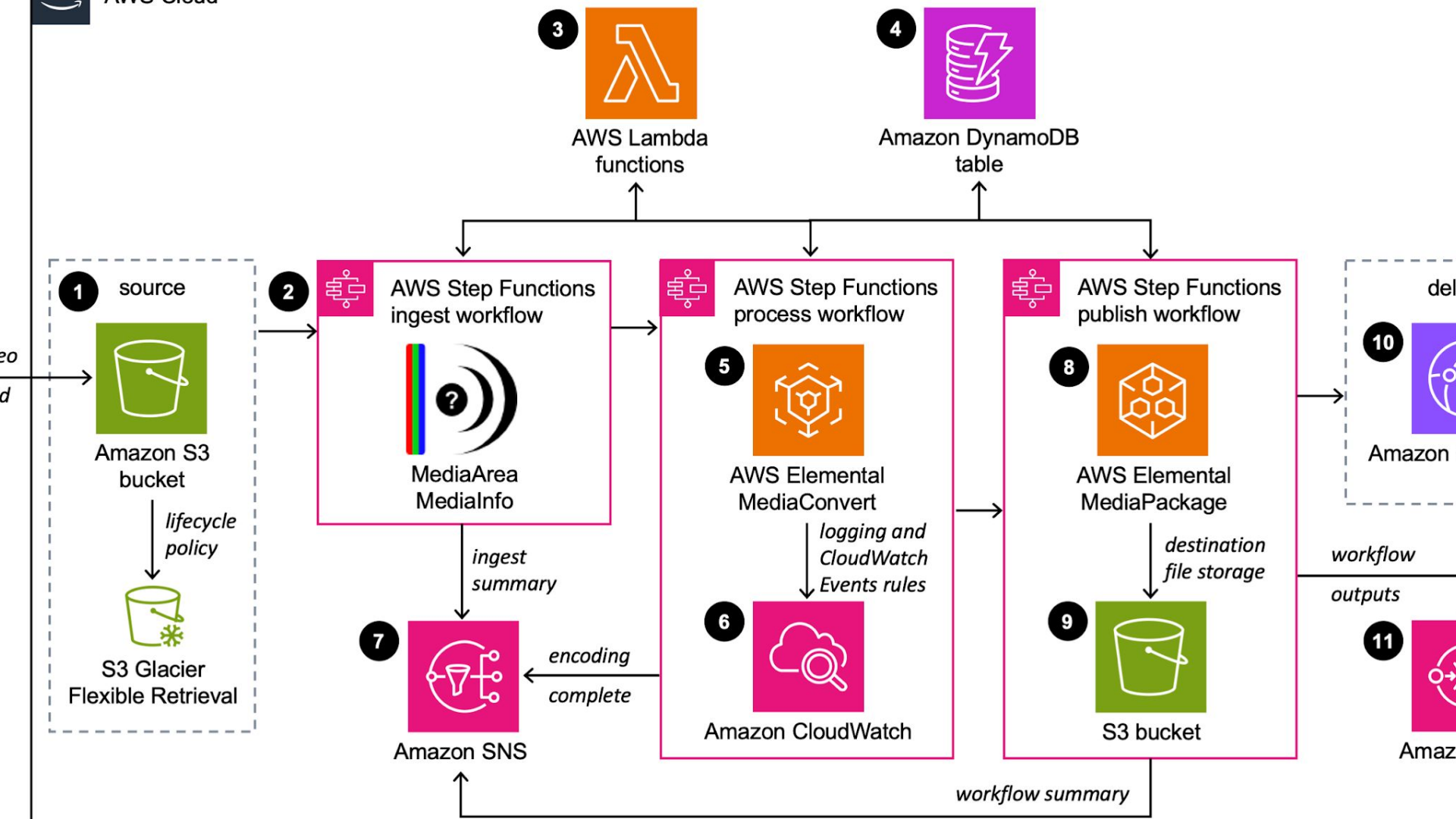
grug smash

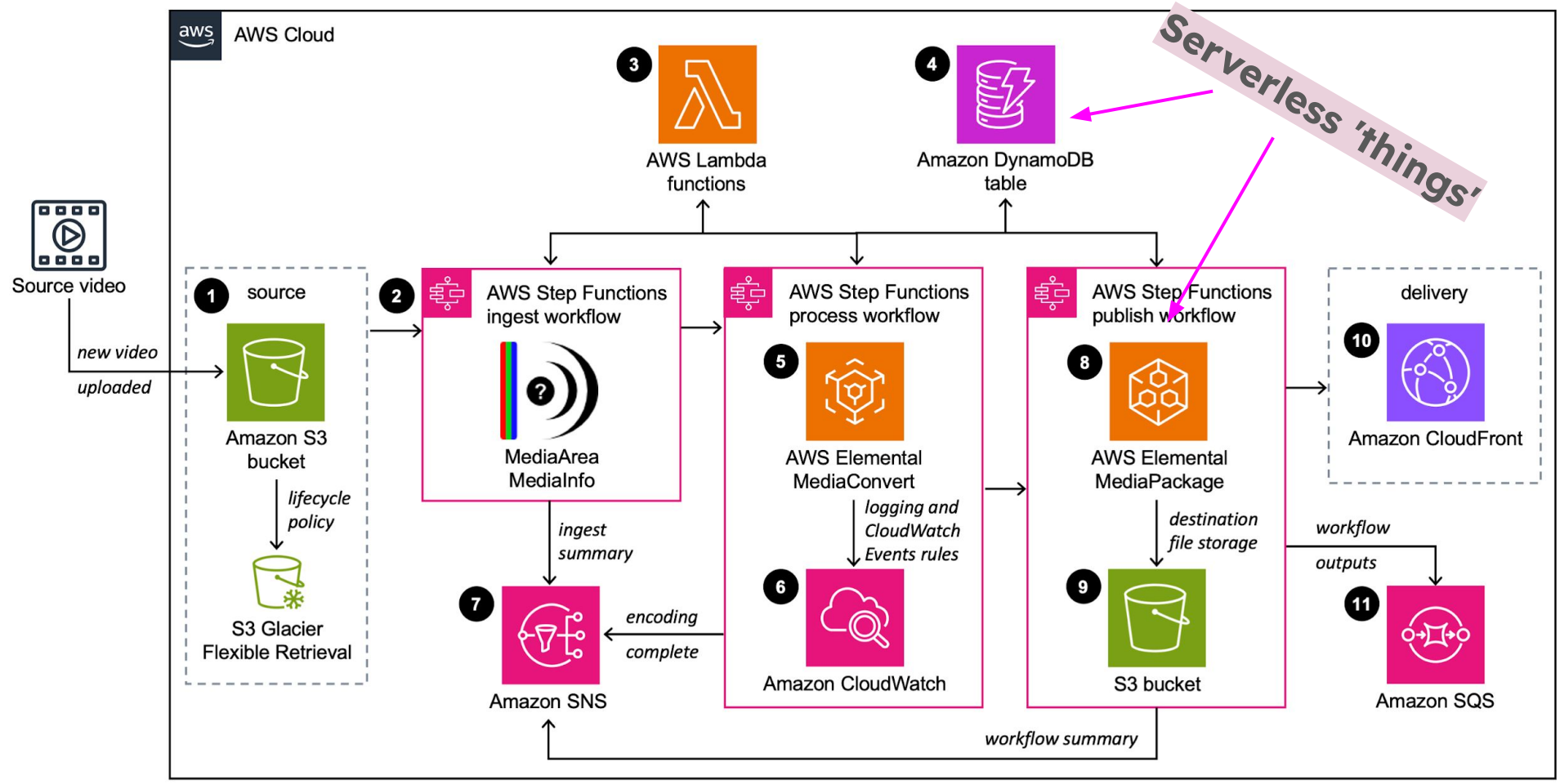


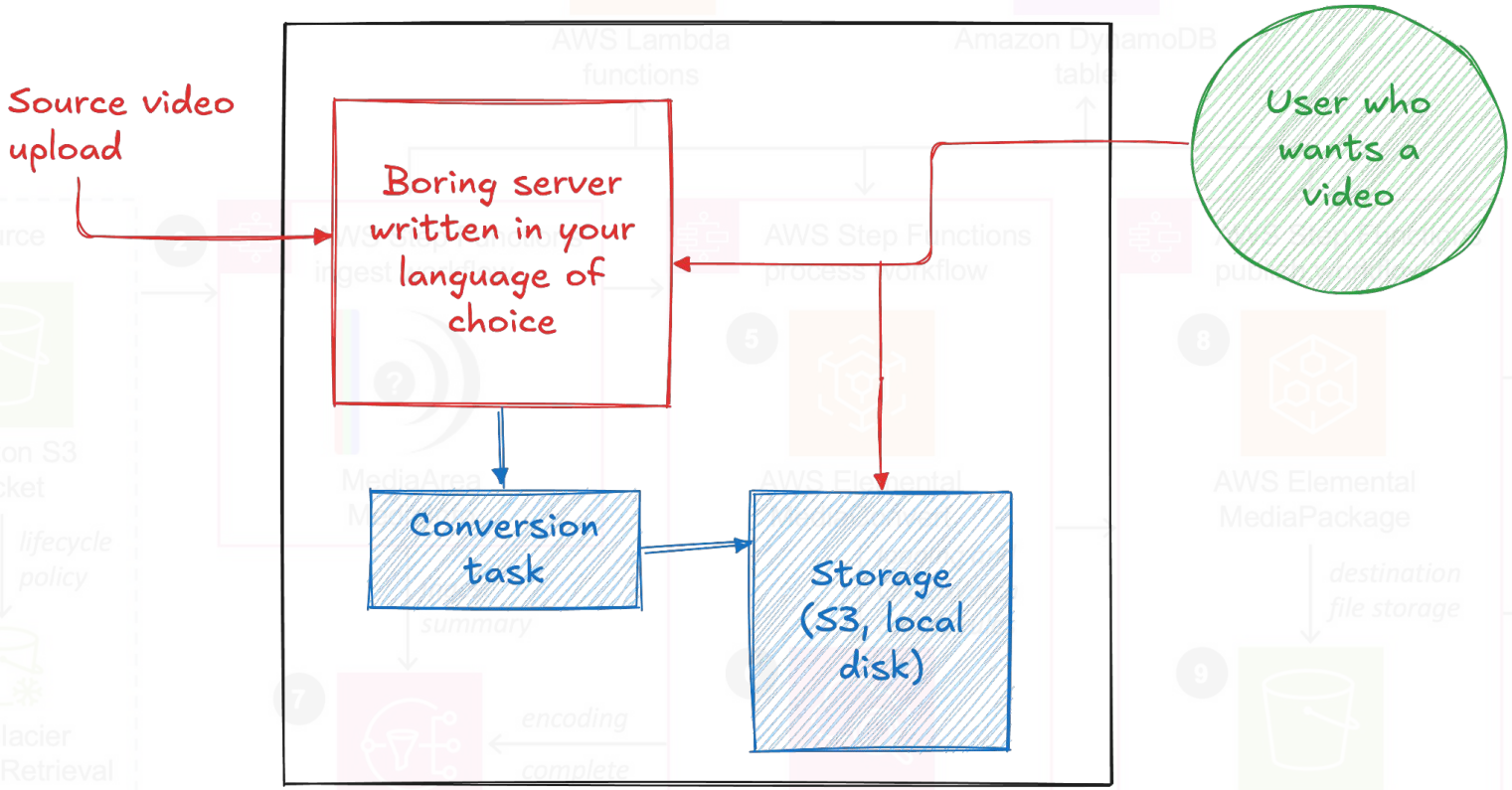
"the cloud"

🐉🐉🐉


Source video
new video
uploaded
AWS Cloud
1
source
Amazon S3
bucket
lifecycle
policy
S3 Glacier
Flexible Retrieval
2
AWS Step Functions
ingest workflow
MediaArea
MediaInfo
ingest
summary
7
Amazon SNS
encoding
complete
AWS Step Functions
process workflow
5
AWS Elemental
MediaConvert
logging and
CloudWatch
Events rules
6
Amazon CloudWatch
delivery
10
Amazon CloudFront
workflow
outputs
11
Amazon SQS

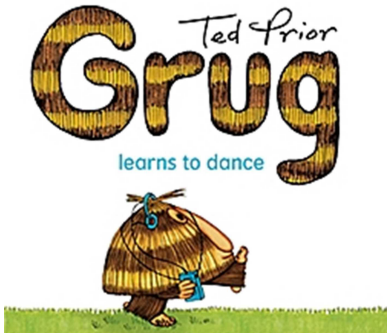

3
AWS Lambda functions
4
Amazon DynamoDB table
1
source
Amazon S3 bucket
lifecycle policy
S3 Glacier Flexible Retrieval
2
AWS Step Functions ingest workflow
MediaArea MediaInfo
ingest summary
7
Amazon SNS
encoding complete
AWS Step Functions process workflow
5
AWS Elemental MediaConvert
logging and CloudWatch Events rules
6
Amazon CloudWatch
AWS Step Functions publish workflow
8
AWS Elemental MediaPackage
destination file storage
9
S3 bucket
workflow summary
10
Amazon
workflow outputs
11

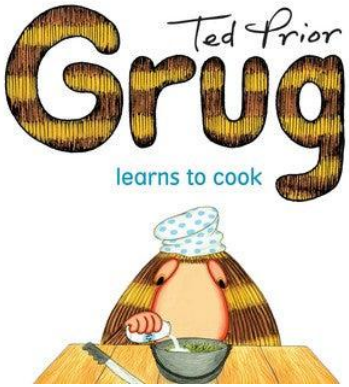

AWS Cloud
Serverless 'things'
3
AWS Lambda functions
4
Amazon DynamoDB table
Source video
new video uploaded
1
source
Amazon S3 bucket
lifecycle policy
S3 Glacier Flexible Retrieval
2
AWS Step Functions ingest workflow
MediaArea MediaInfo
ingest summary
AWS Step Functions process workflow
5
AWS Elemental MediaConvert
logging and CloudWatch Events rules
6
Amazon CloudWatch
AWS Step Functions publish workflow
8
AWS Elemental MediaPackage
destination file storage
9
S3 bucket
delivery
10
Amazon CloudFront
workflow outputs
11
Amazon SQS
7
Amazon SNS
encoding complete
workflow summary

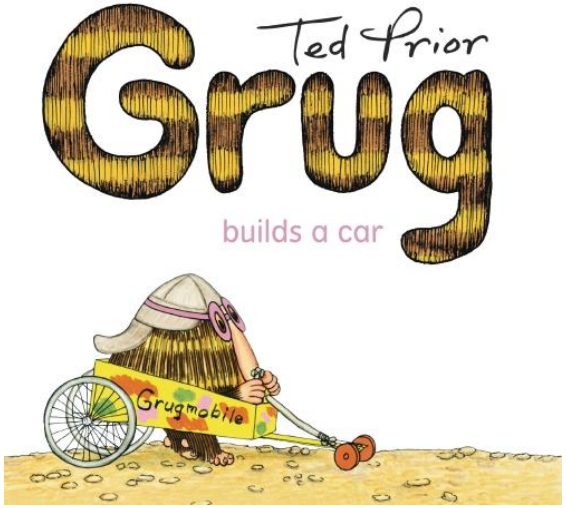

Source video upload
Boring server written in your language of choice
User who wants a video
Conversion task
Storage (S3, local disk)

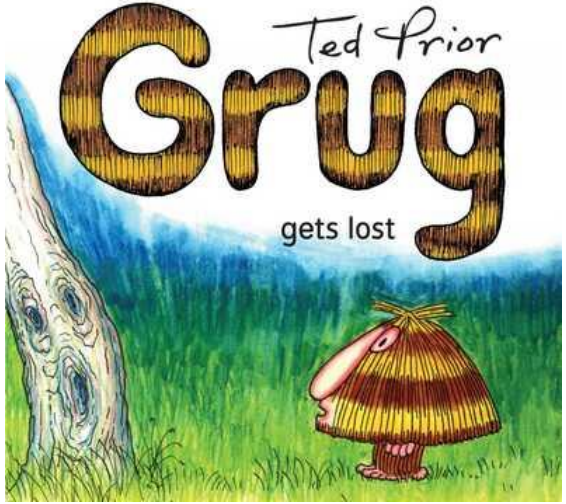
The "Well-Lit Path"

"grug brained developer"







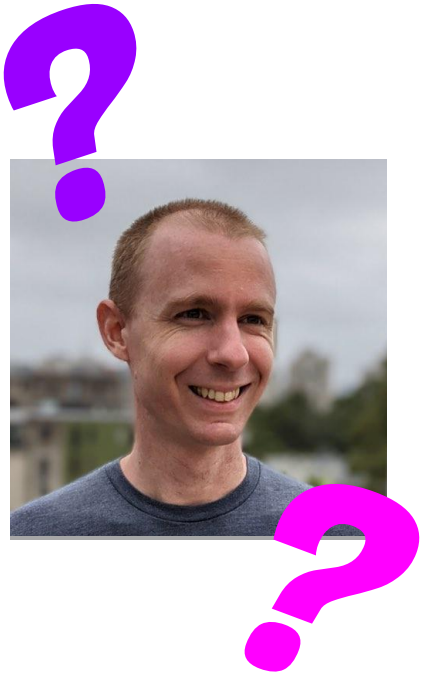


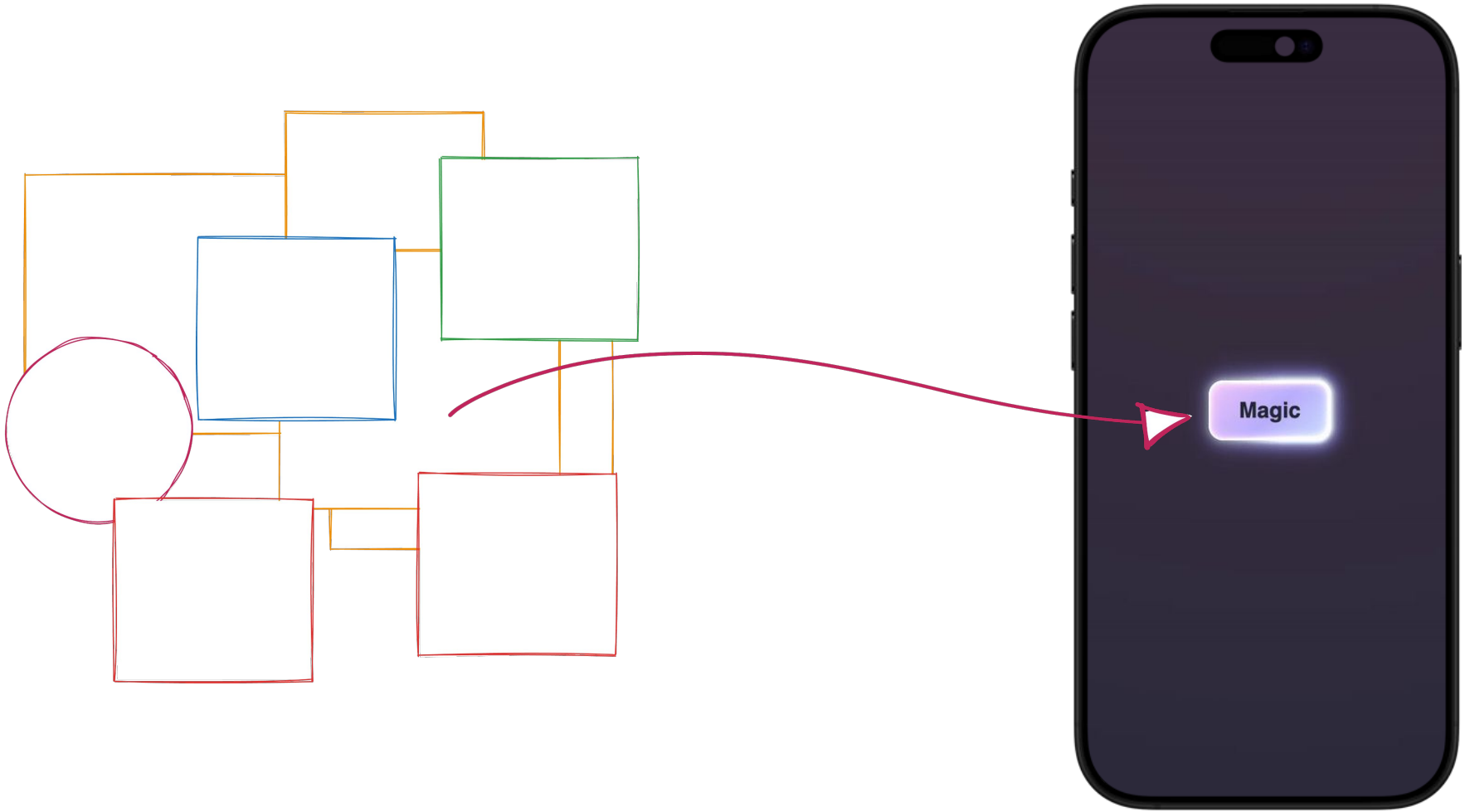
Your takeaways...

Philosophy 🧘

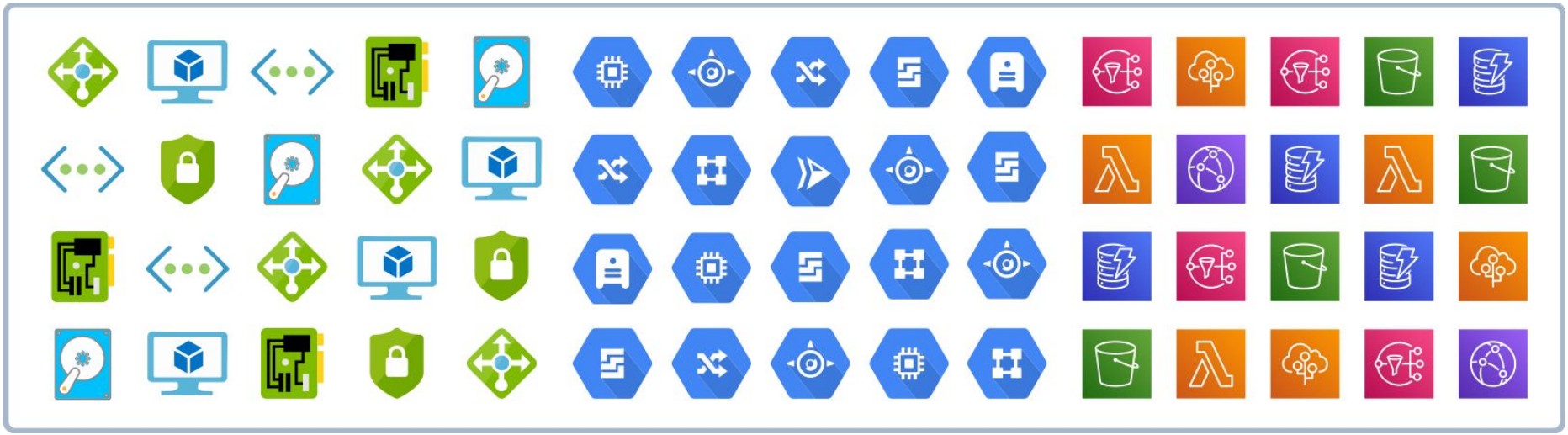
🕸 Complexity

Be better-er ✊

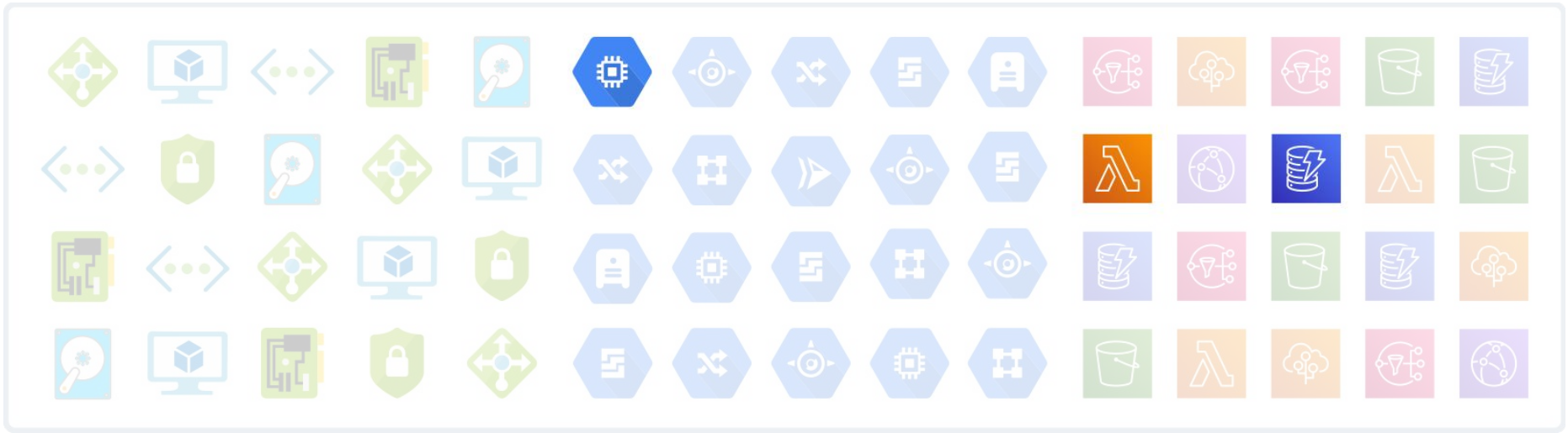



Magic

Primitive overload



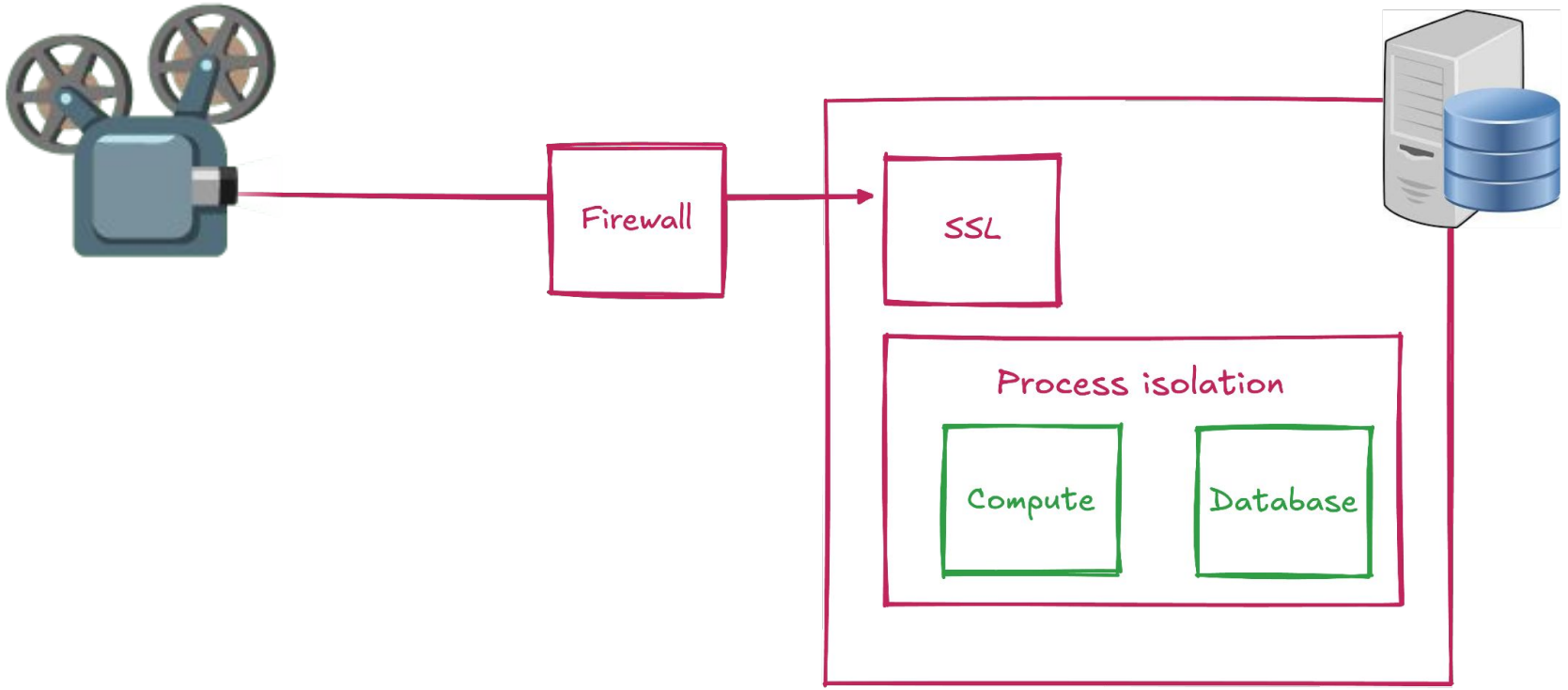
Primitive overload

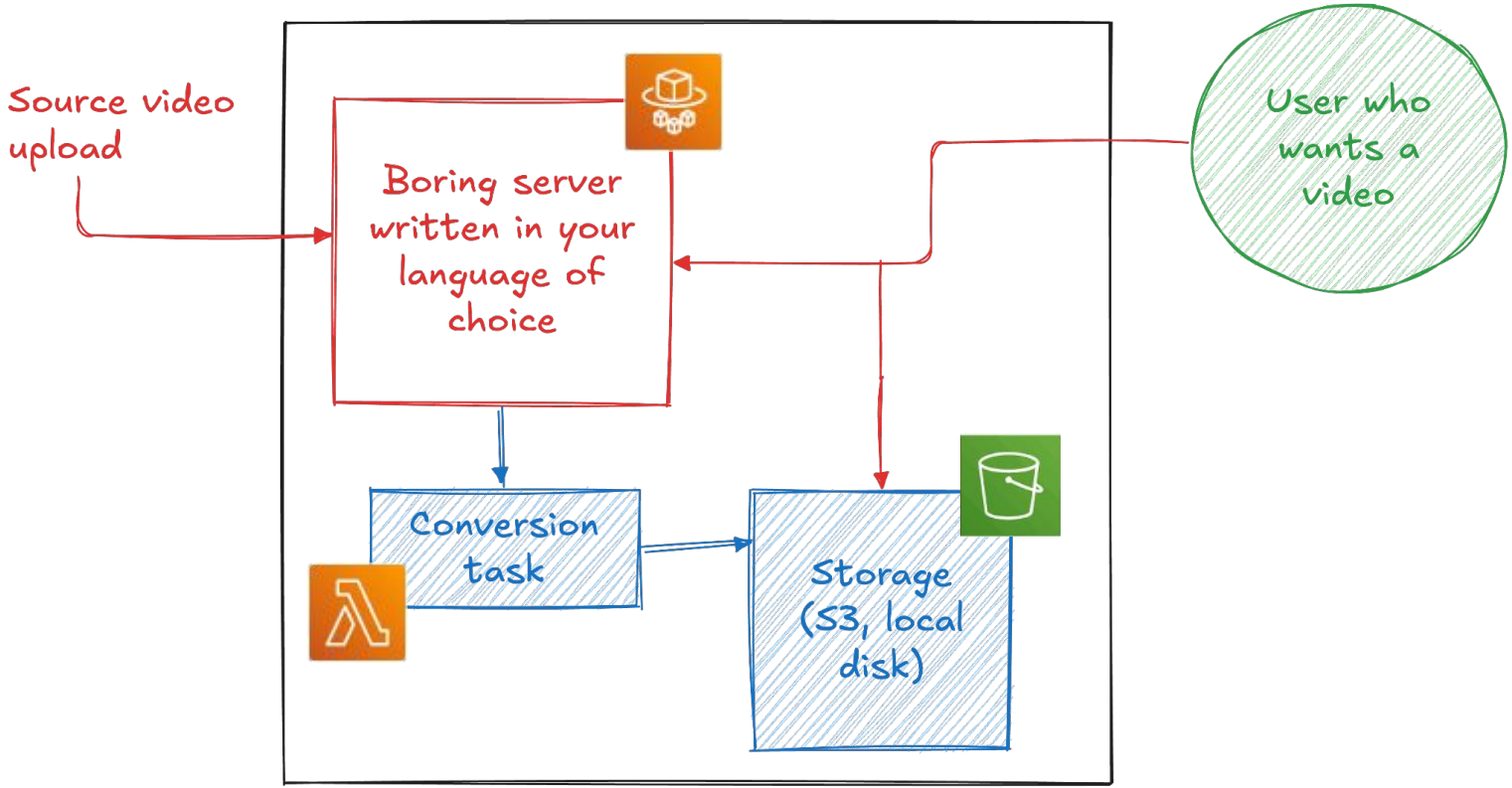


Gear
Big Wallet
We're good at "this"
R
ZR
Pouch
8
8
16
+900
Options
Back B

Gear
Big Wallet
We're good at "this"
R
ZR
R
Pouch
Options
Back B

The alternative?


Firewall
SSL
Process isolation
Compute
Database


Source video upload
Boring server written in your language of choice
User who wants a video
Conversion task
Storage (S3, local disk)

Pattern rundown

🚫 Don't: Build the MVP of being online

- If you find yourself…
  - Getting an IP address
  - Issuing SSL certs
  - Reverse proxying from front-ends to your HTTP server
  - Worrying about DDOS attacks
- Maybe find a provider that handles this for you
- You should be able to host boring HTTP and have it show up

✅ Do: Serve HTTP requests

- Your language of choice has a way to serve HTTP
  - HTTP has 'won'; don't talk some weird protocol!
- If long-lived connections are hard - you're not serving HTTP
  - e.g., WebSocket

✅ Do: Get things talking to each other

- Cloud configs give you the ability to "wire" things together
  - But at the cost of complexity
- What is your actual goal?
  - Get services conceptually on the same LAN, same machine, … etc
  - Same as the MVP—avoid the overheads of using things
- Alternative: Tailscale
  - "The New Internet" 🔍
  - Just puts devices on a local LAN - trivial/zero-cost VPC

✅ Do: Prevent unauthorized access

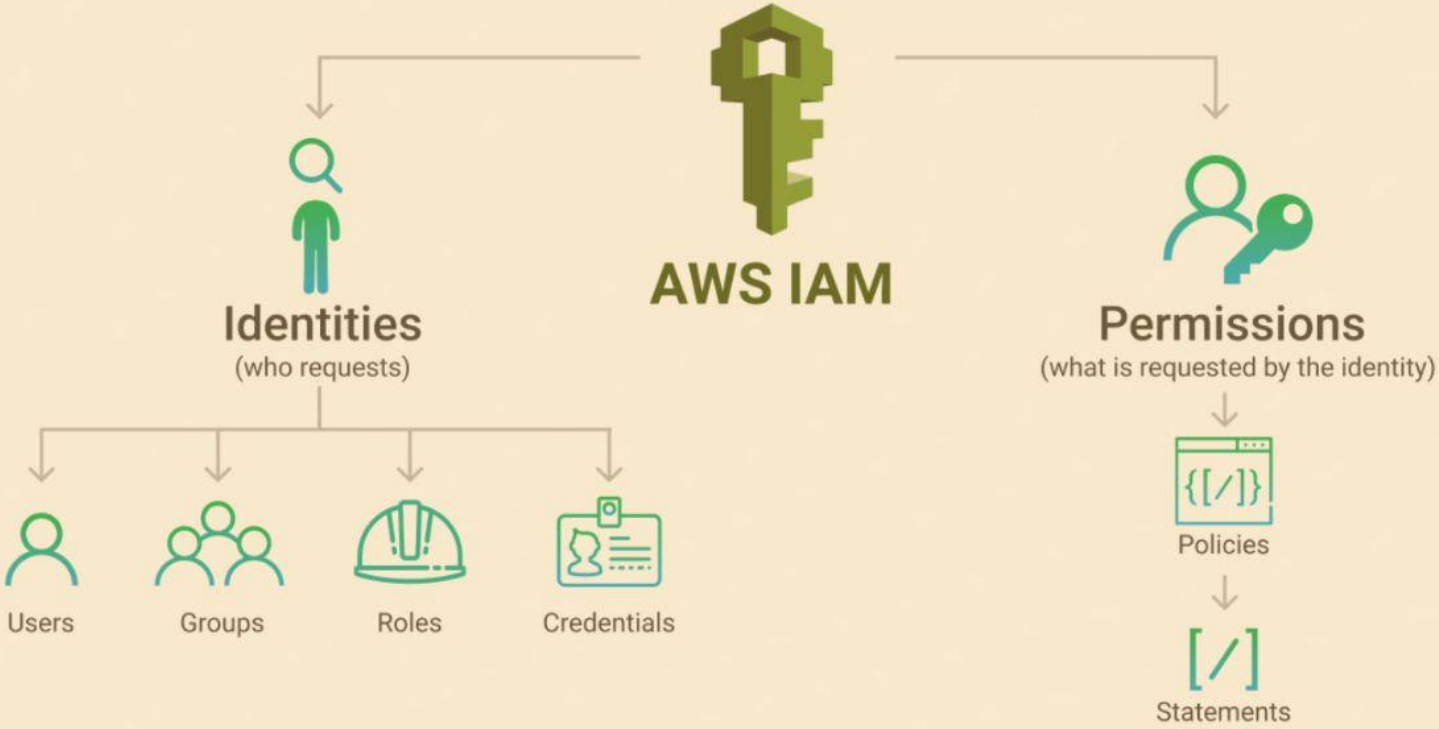
- Corollary: allow authorized access with the right permissions
  - Includes user access, service access, …
- Alternative: uhhhh You do need OAuth/JWTs, I'm sorry
  - Maybe don't roll your own

🚫 Don't: Build complex solutions for scale

- Serverless is often touted as a way to 'scale infinitely'
  - Your bills?
- "Good problem to have"
- Solve it low-tech
  - Horizontal sharding - your MVP of being online can handle this
  - Read-only replicas, too busy notices
  - "1000 people in the queue"

🤔 Aside: Unknowns

- I want to stop bad actors / unknown exploits / etc
  - Unknown unknowns of your system
- What can help you prevent things you don't know about? 🤔
  - Less is more


AWS IAM
Identities
(who requests)
Users
Groups
Roles
Credentials
Permissions
(what is requested by the identity)
Policies
Statements

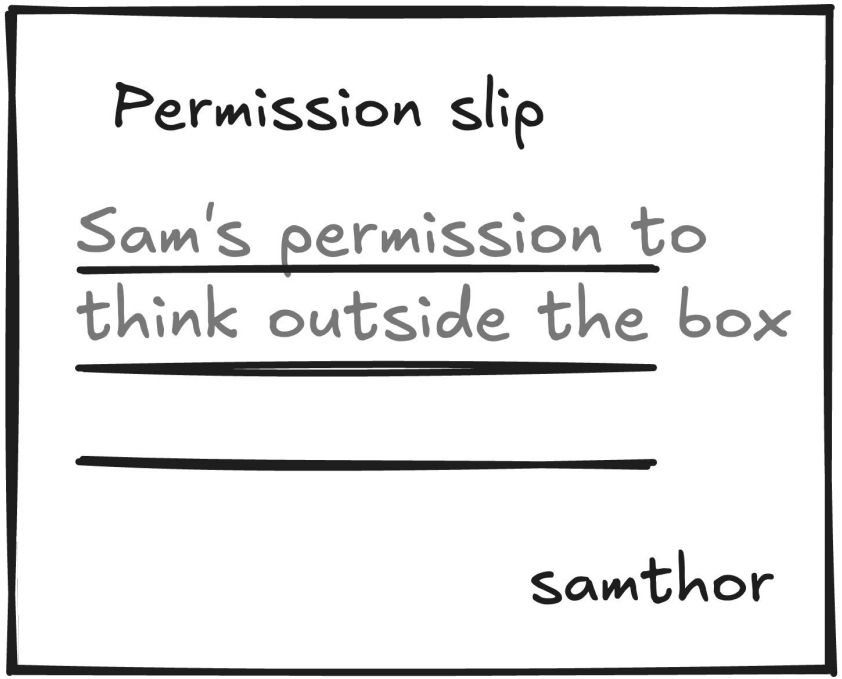
Hello there. I'm Sam, and I write about the web. I'm available for consulting. See some open-source packages I've built.

Looking for older posts? Check out the archive. You can find me on Mastodon or Xwitter. Add my blog to your feed reader. 📚

Popular Posts

Node.js can run TypeScript October 2024
Rejoyce through a combination of command-line flags

AWS Amplify Is A Grift March 2023
Something something Amplify is bad

Observing rendered DOM nodes January 2021
Observe DOM appearance, size and and rendering position on a page.

It's dangerous to go alone!
Take this