TURBINE LABS

Our Move to Envoy

Replacing NGINX with Envoy in a Traffic Control System

Kubecon EU May 2nd 2018

TURBINE LABS

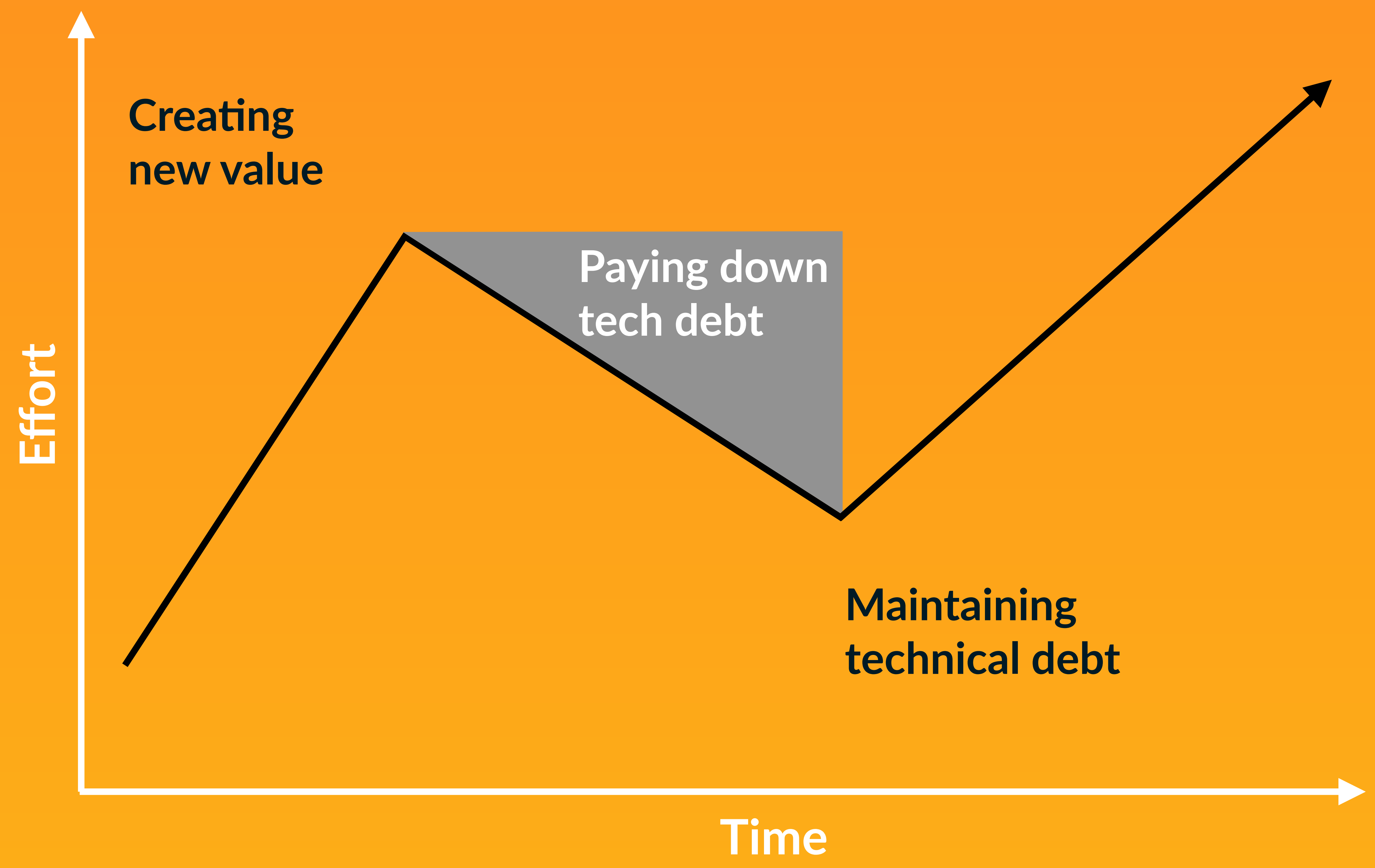
Mark

@mccv on Twitter

- **Twitter** — platform/API team during the great de-monolithing, traffic management pioneer
- **Nest** — led service engineering, launched their developer program
- **Turbine Labs** — traffic management superpowers for everyone

They’re the sole scalable fix to tech debt

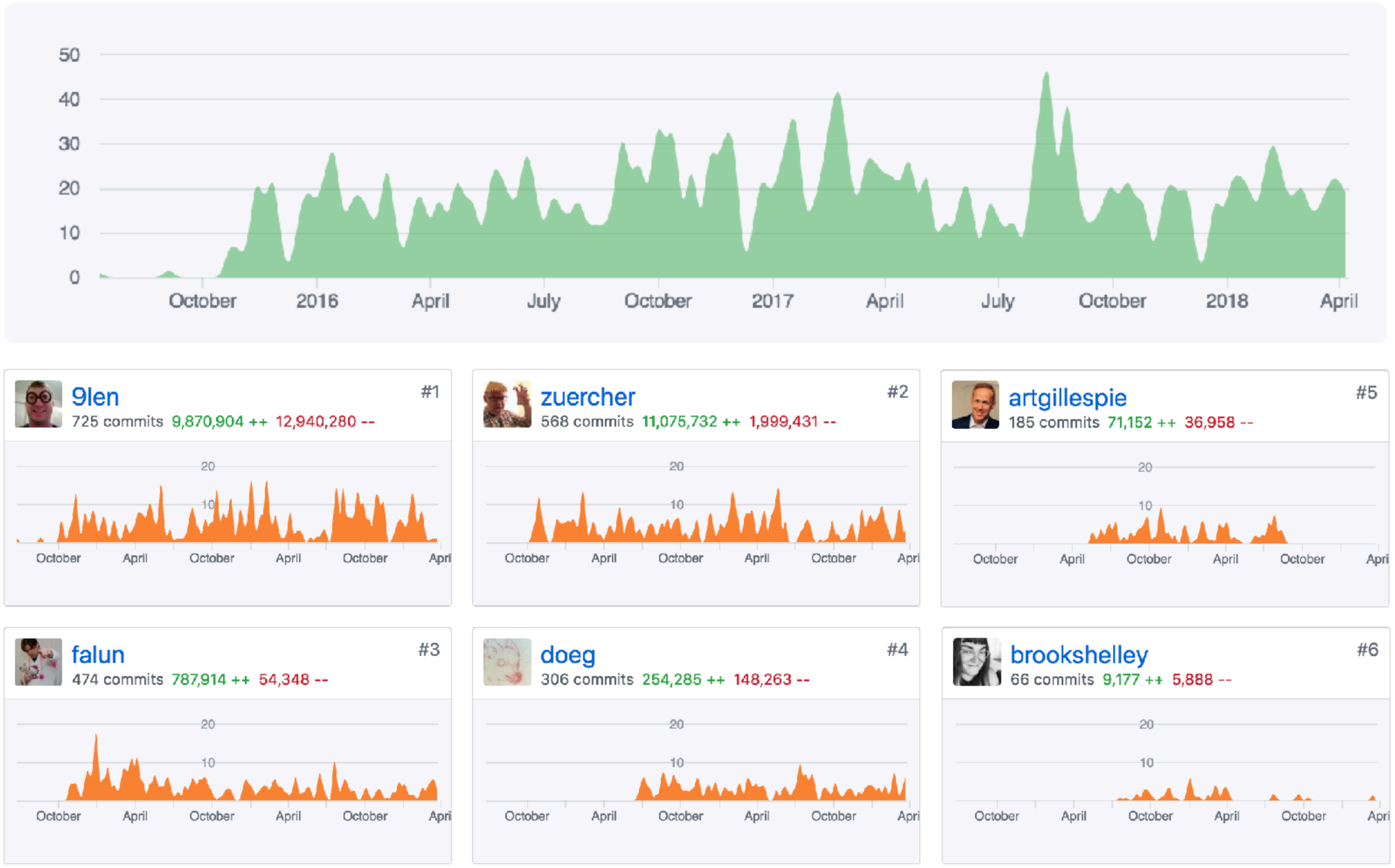
Will Larson, https://lethain.com/migrations/



The Case for Migrations

- Most tools and processes support about an order of magnitude of growth.
- Migrating to a new system is the path to getting the next order of magnitude.
- Let’s make your Envoy migration easier.
- Then let’s make more migrations easier.

Disclaimer: I did almost none of this work



9len #1
725 commits 9,870,904 ++ 12,940,280 --

zuercher #2
568 commits 11,075,732 ++ 1,999,431 --

artgillespie #5
185 commits 71,152 ++ 36,958 --

falun #3
474 commits 787,914 ++ 54,348 --

doeg #4
306 commits 254,285 ++ 148,263 --

brookshelley #6
66 commits 9,177 ++ 5,888 --

Contributions to master, excluding merge comments

TURBINE LABS

Turbine Labs

We build an application that unlocks traffic management superpowers for everyone.

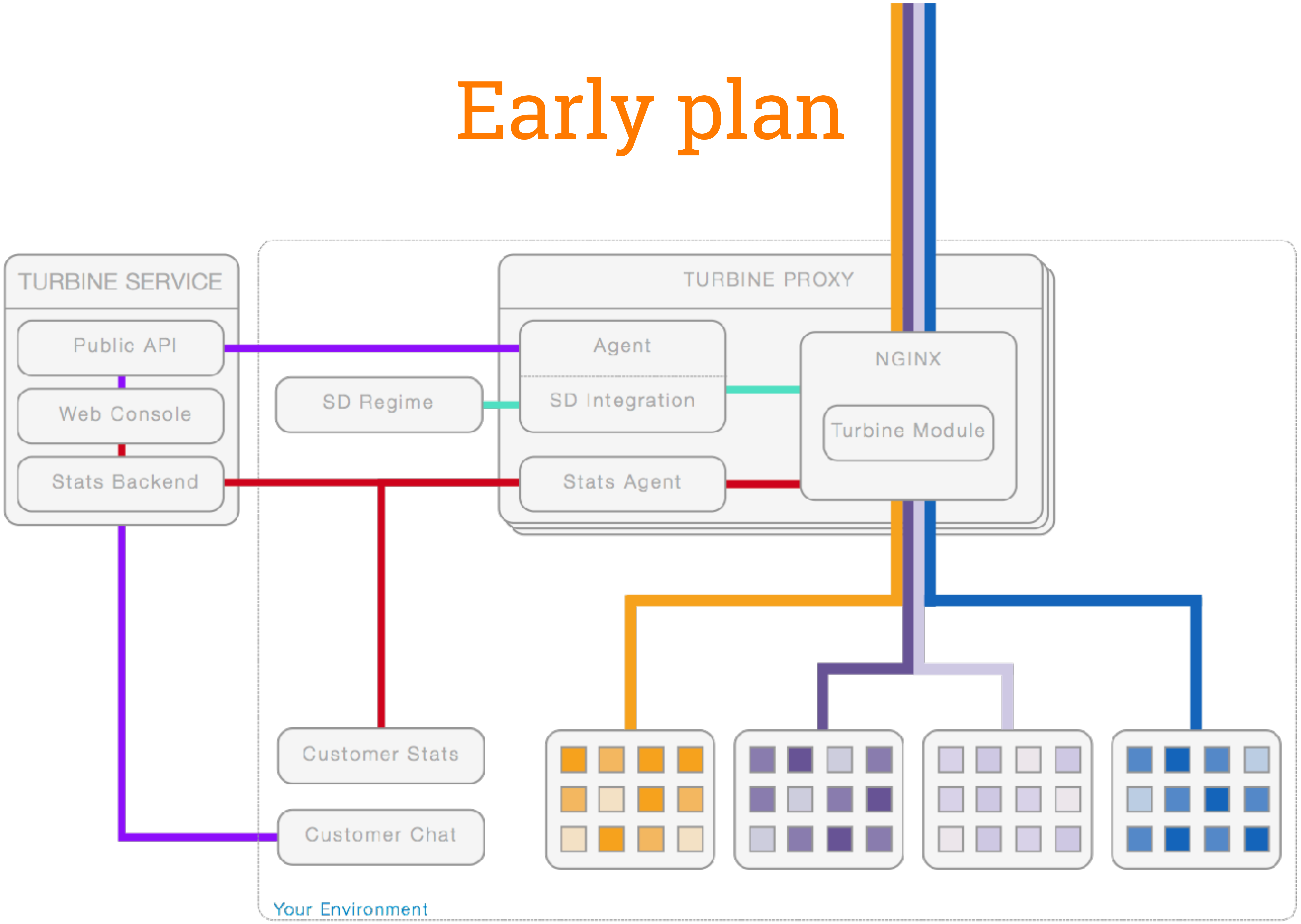
Proxies are at the core of this application.

We are in a unique position as both a vendor and consumer of proxy tech.

Pondering a Change

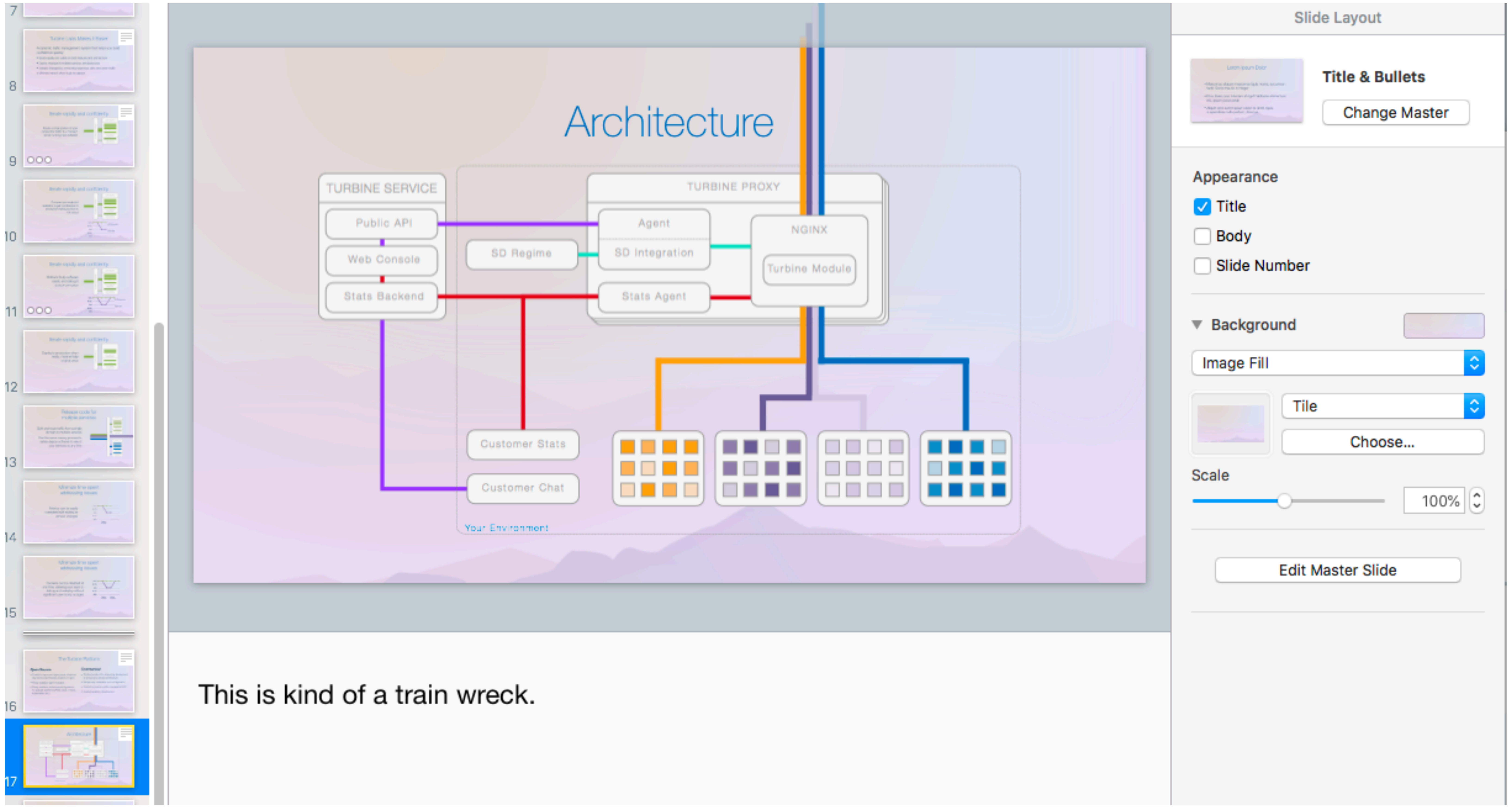
TURBINE LABS

Early plan



TURBINE LABS

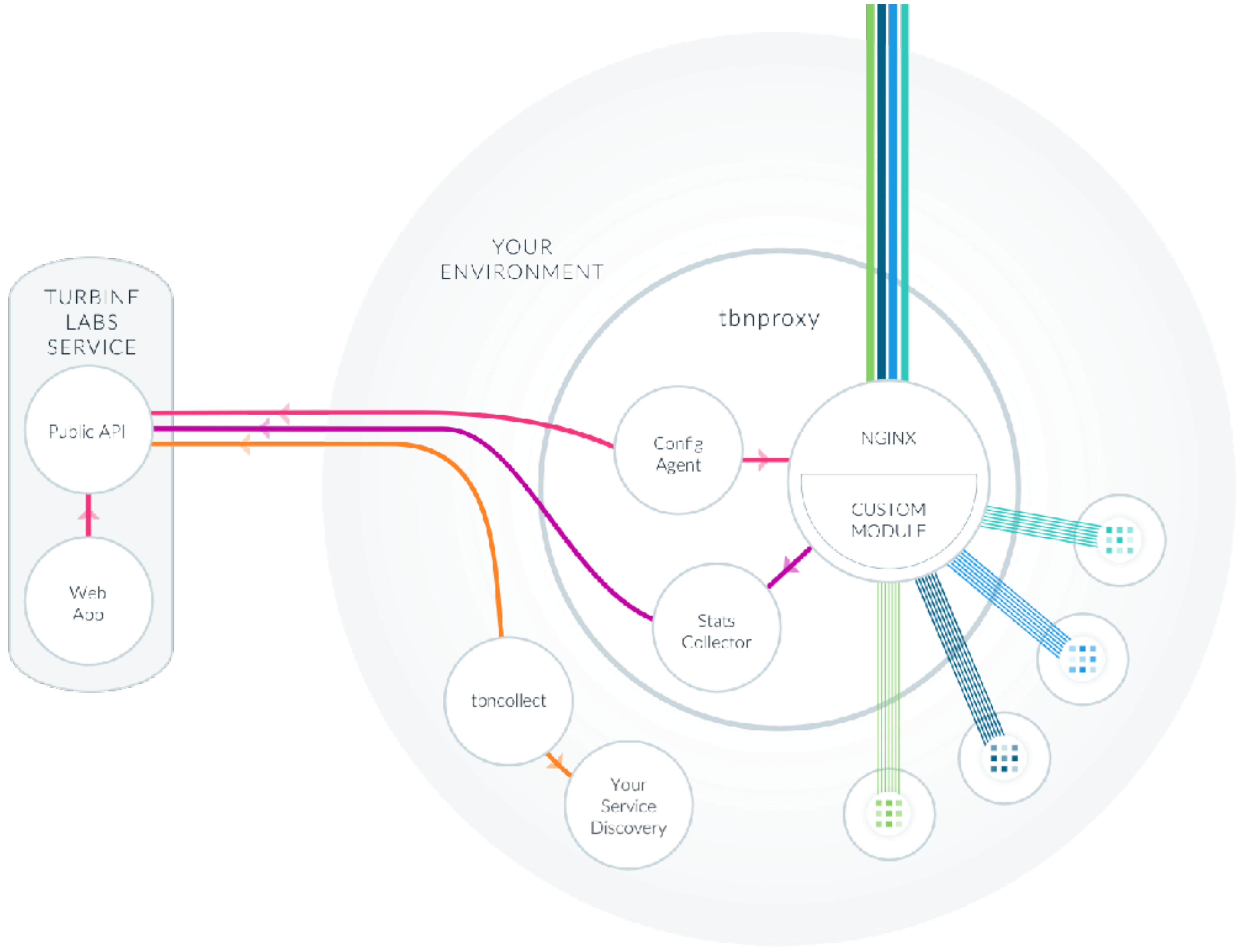
Thoughts on the early plan



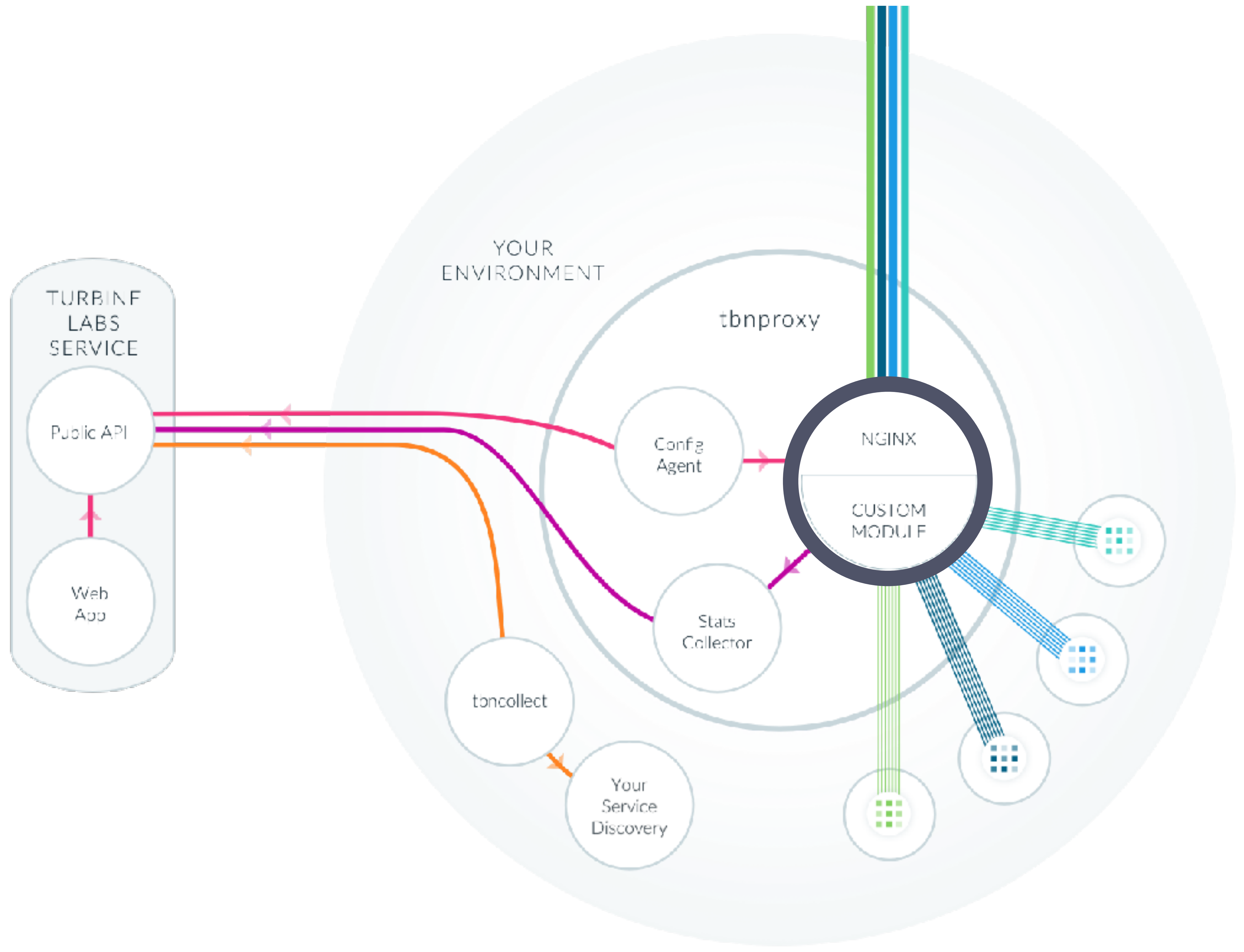
This is kind of a train wreck.

TURBINE LABS

Our initial product



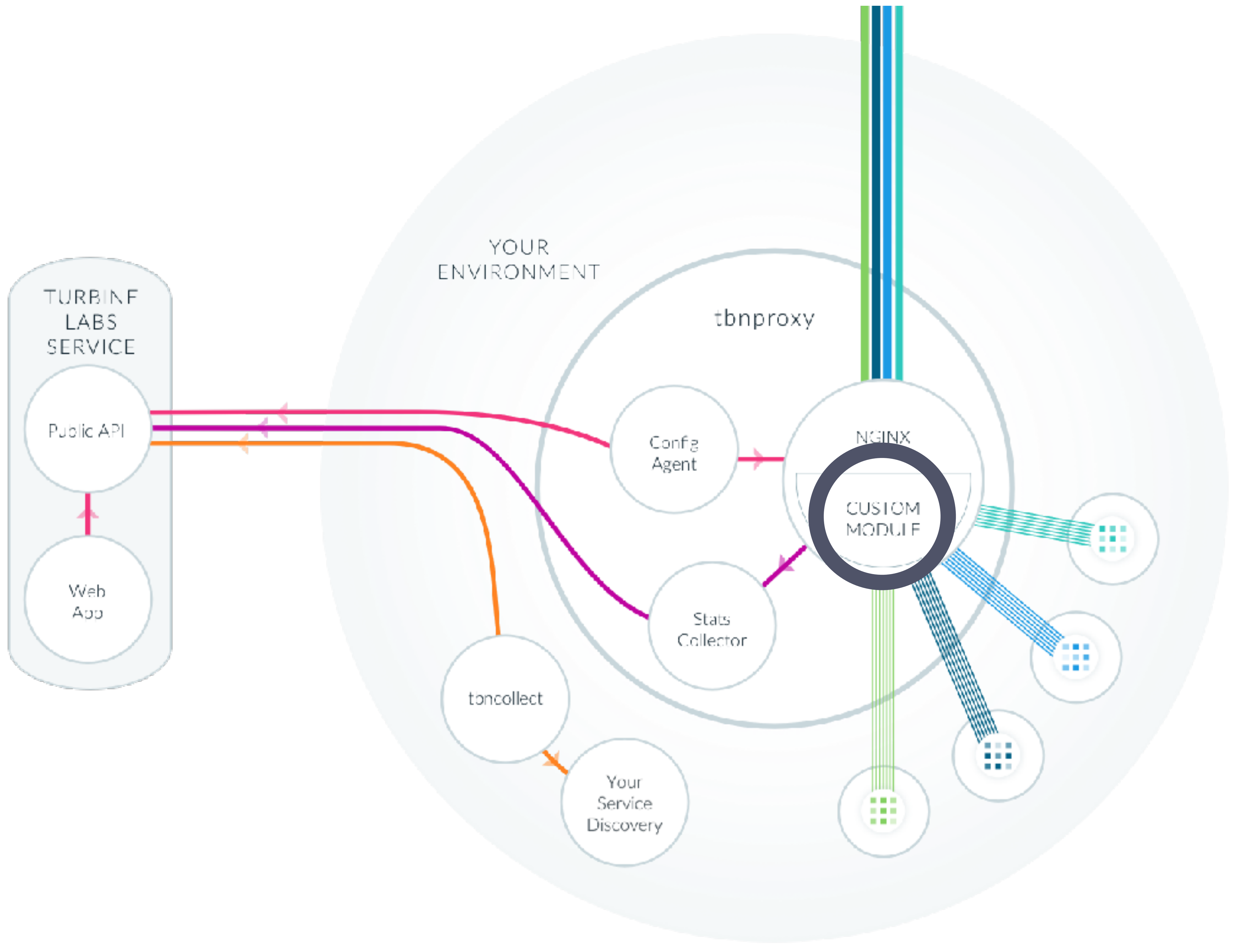
This worked, so why move?



- The base NGINX routing primitives weren't sufficient for our needs.
- Extending NGINX means shipping a custom module.

TURBINE LABS

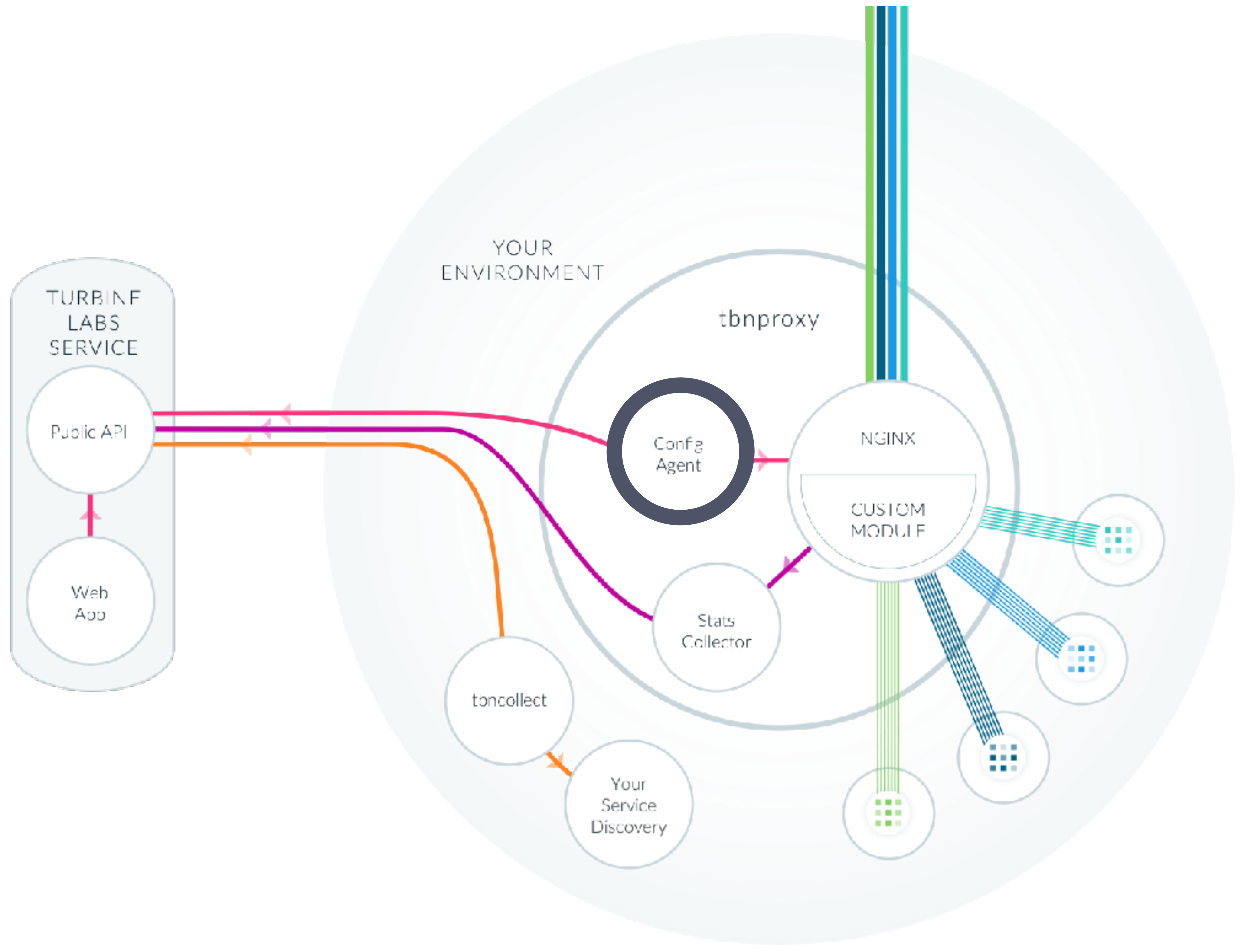
This worked, so why move?



- NGINX's upstream constructs didn't meet our needs either.
- To manage traffic effectively you have to know more about your nodes than their IP and port.

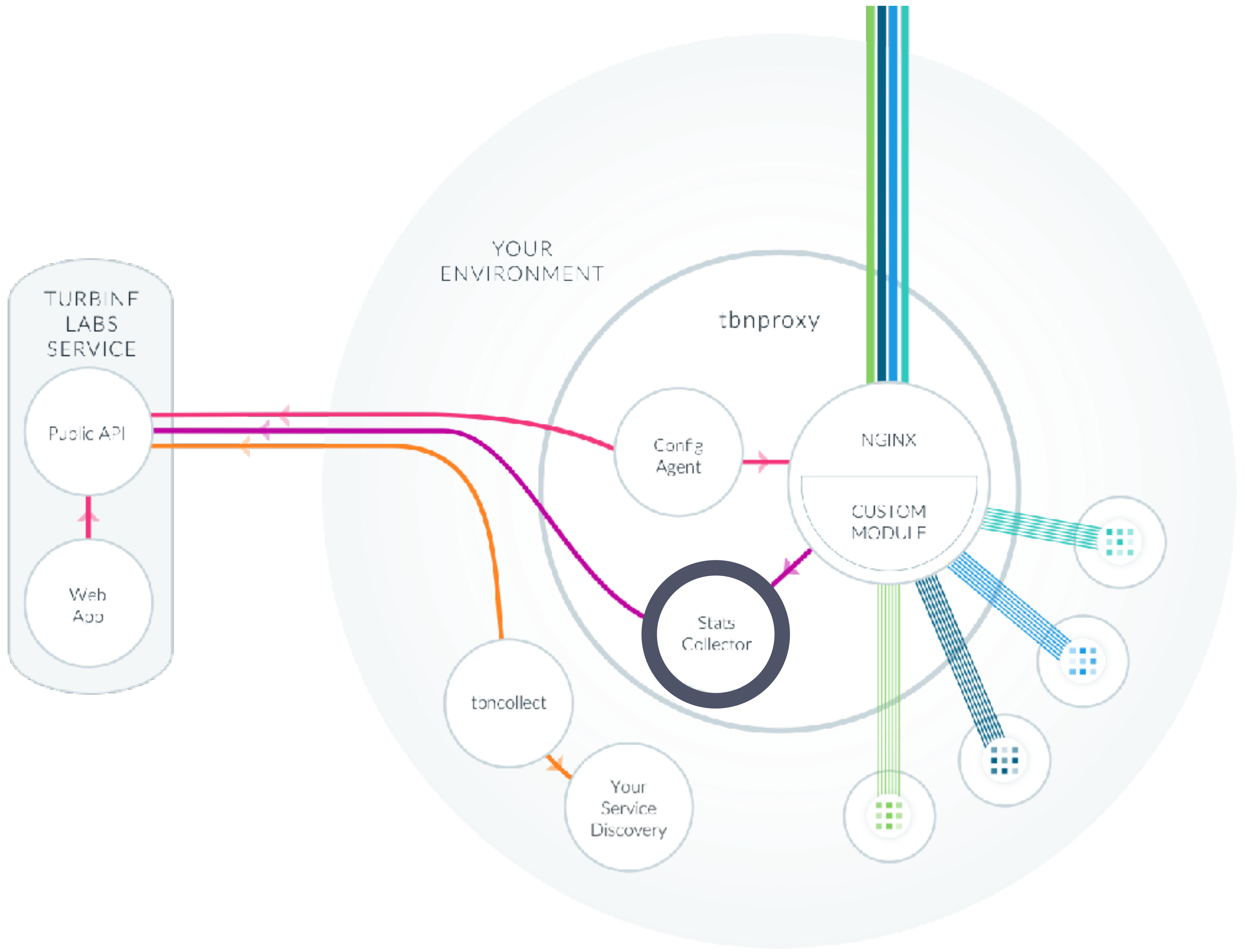
TURBINE LABS

This worked, so why move?



- File-based configuration requires an agent running alongside the NGINX process.
- NGINX reloads are pretty good, but config changes still require a reload.

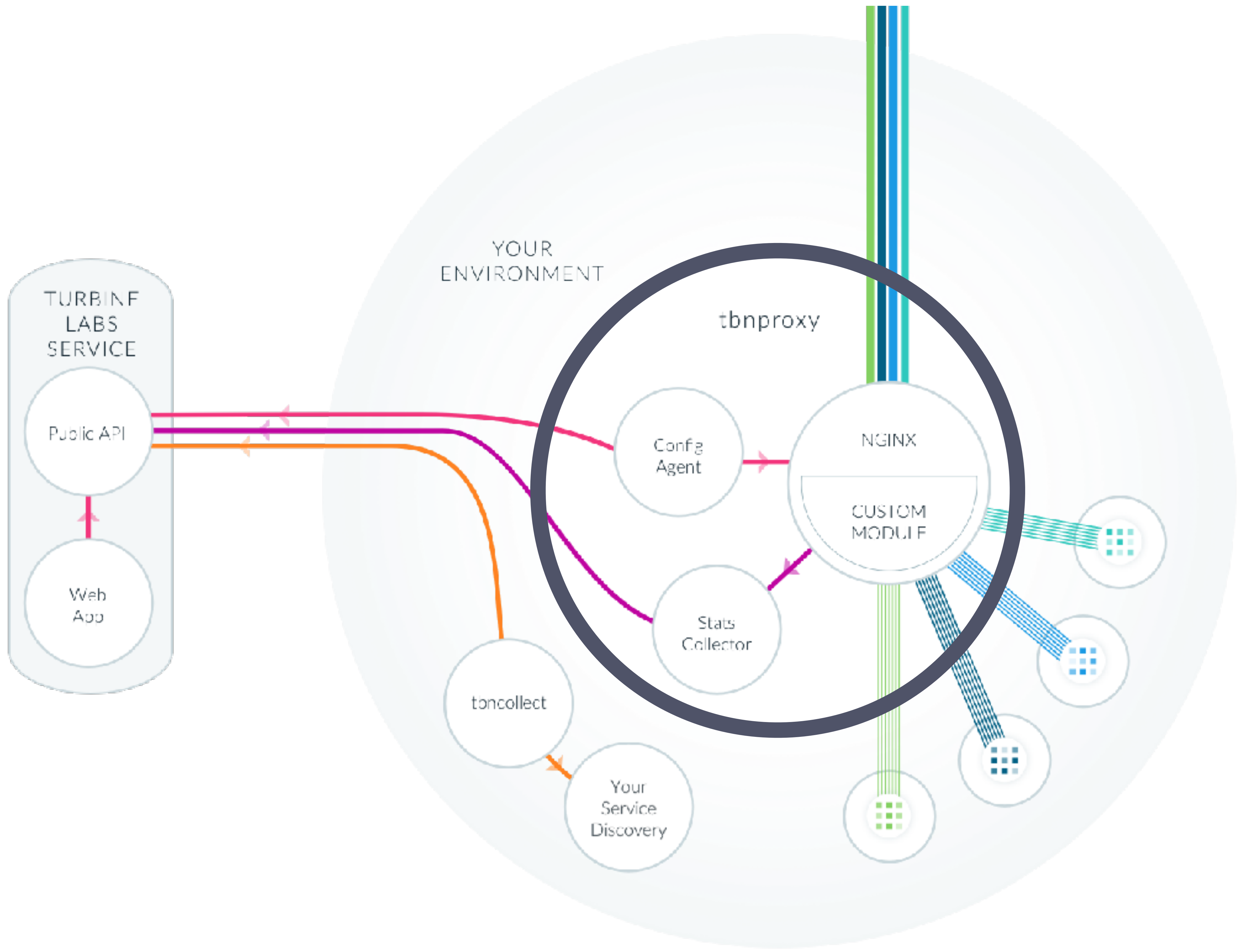
This worked, so why move?



- To get the good stats you need to parse log files.
- Running a parser means colocating a process with the NGINX process.

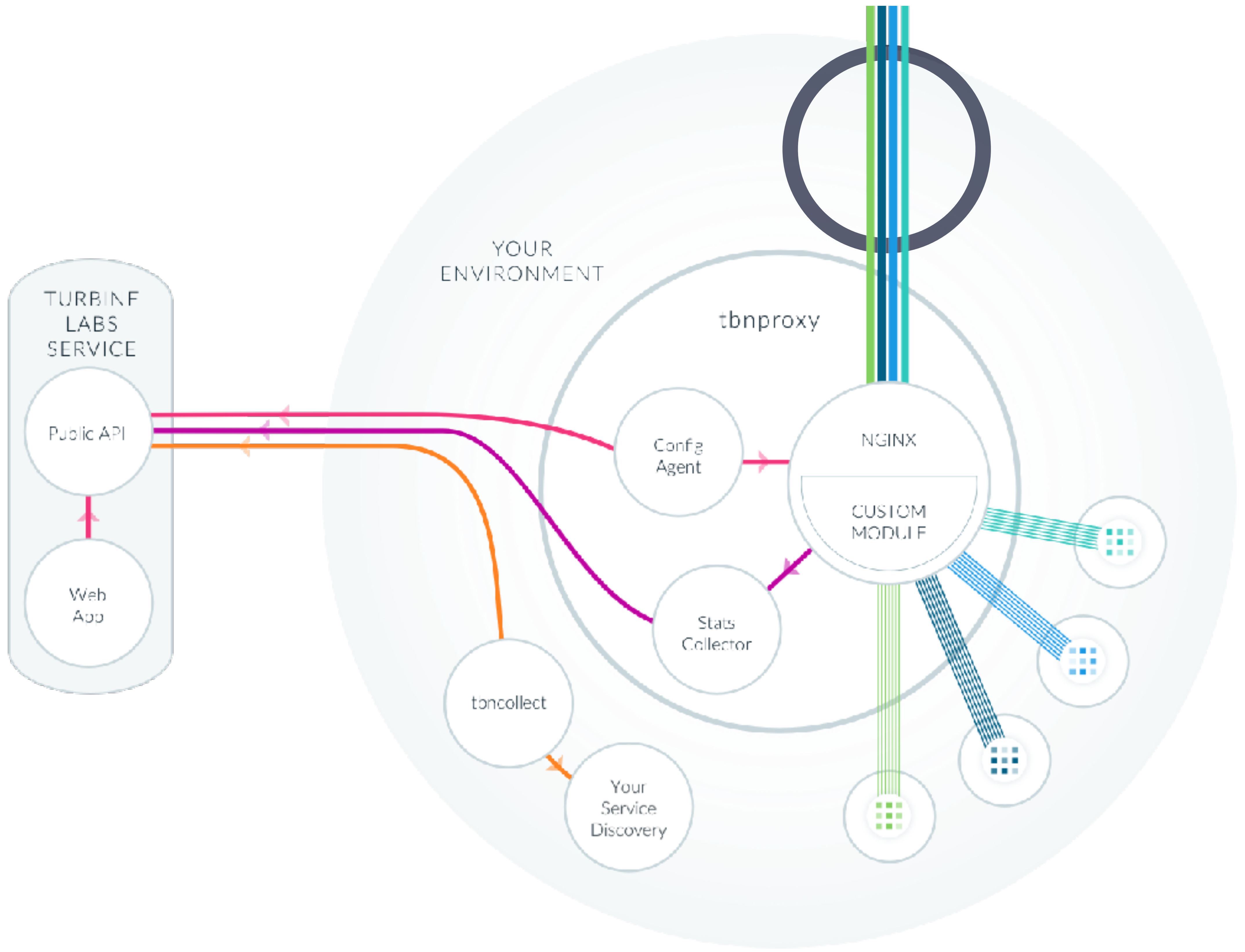
TURBINE LABS

This worked, so why move?

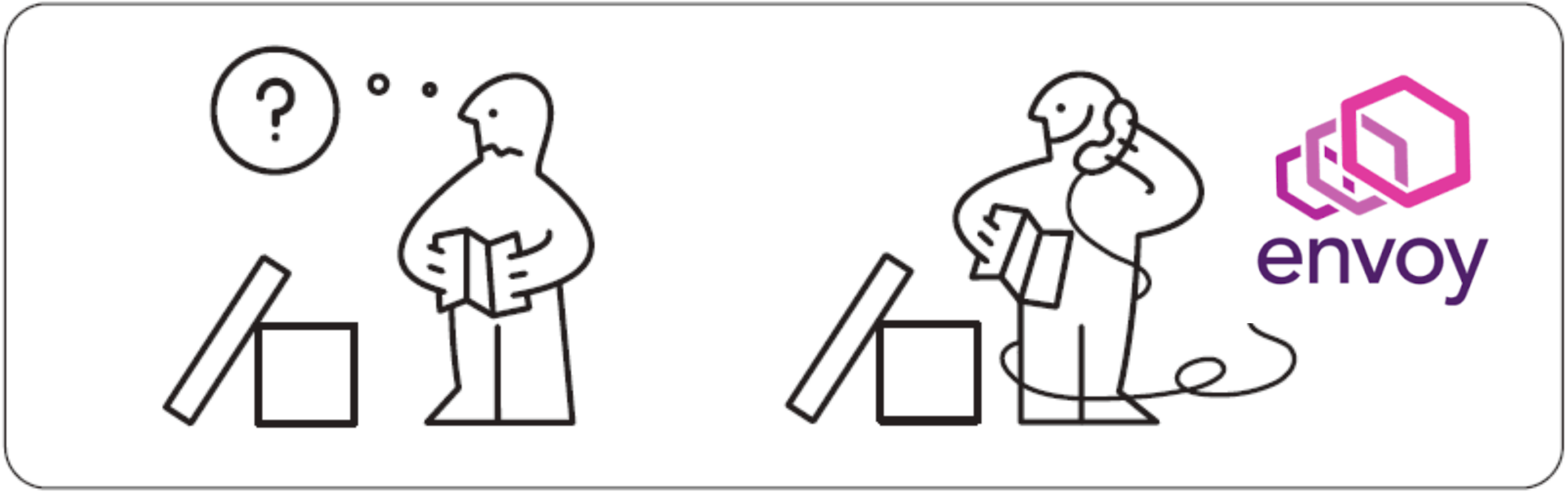


Running agents and a custom module pushed us to ship our own packaged ball of stuff for customers to deploy everywhere.

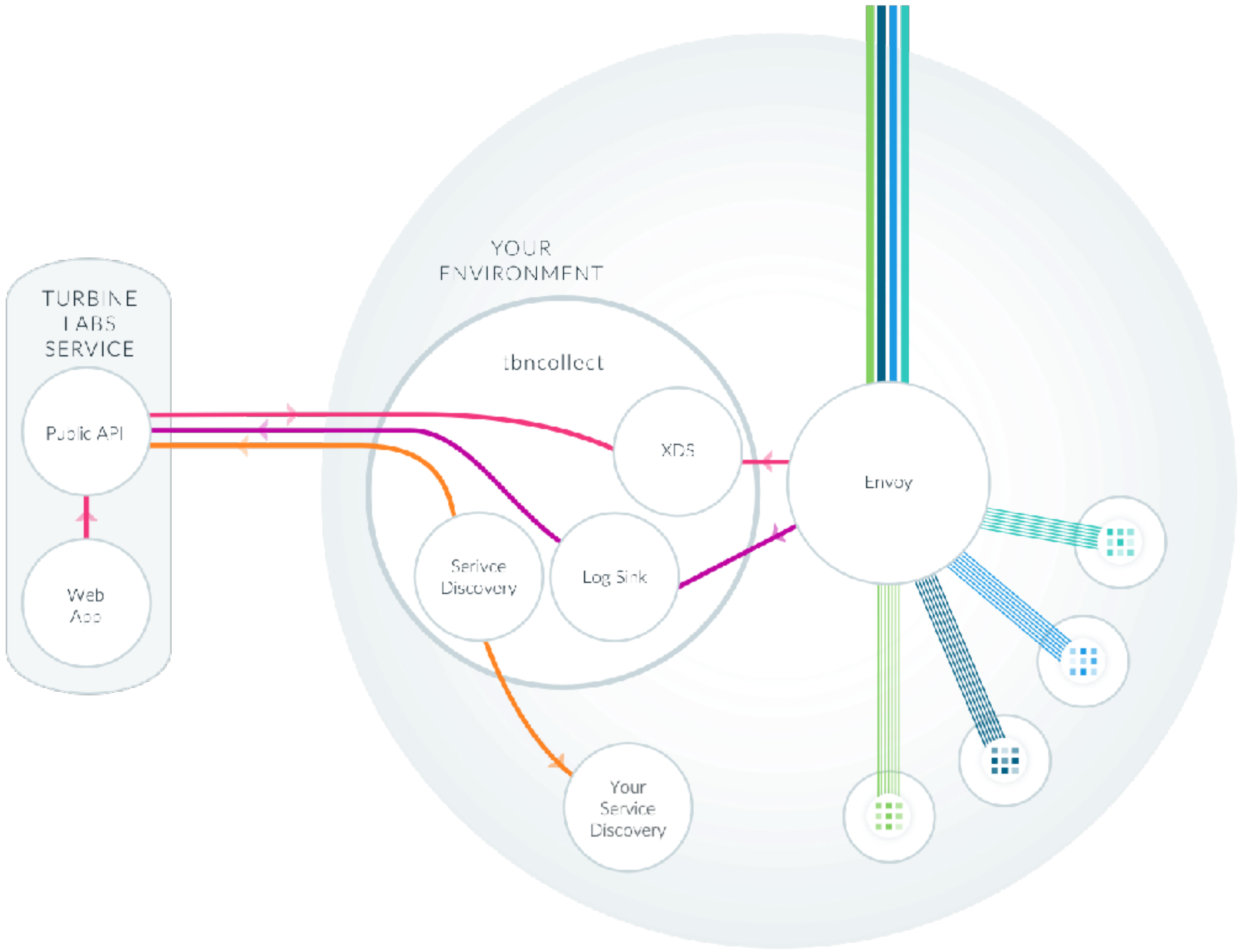
This worked, so why move?



Your protocol options are TCP, HTTP/1[.1], and until very recently only sort of HTTP/2

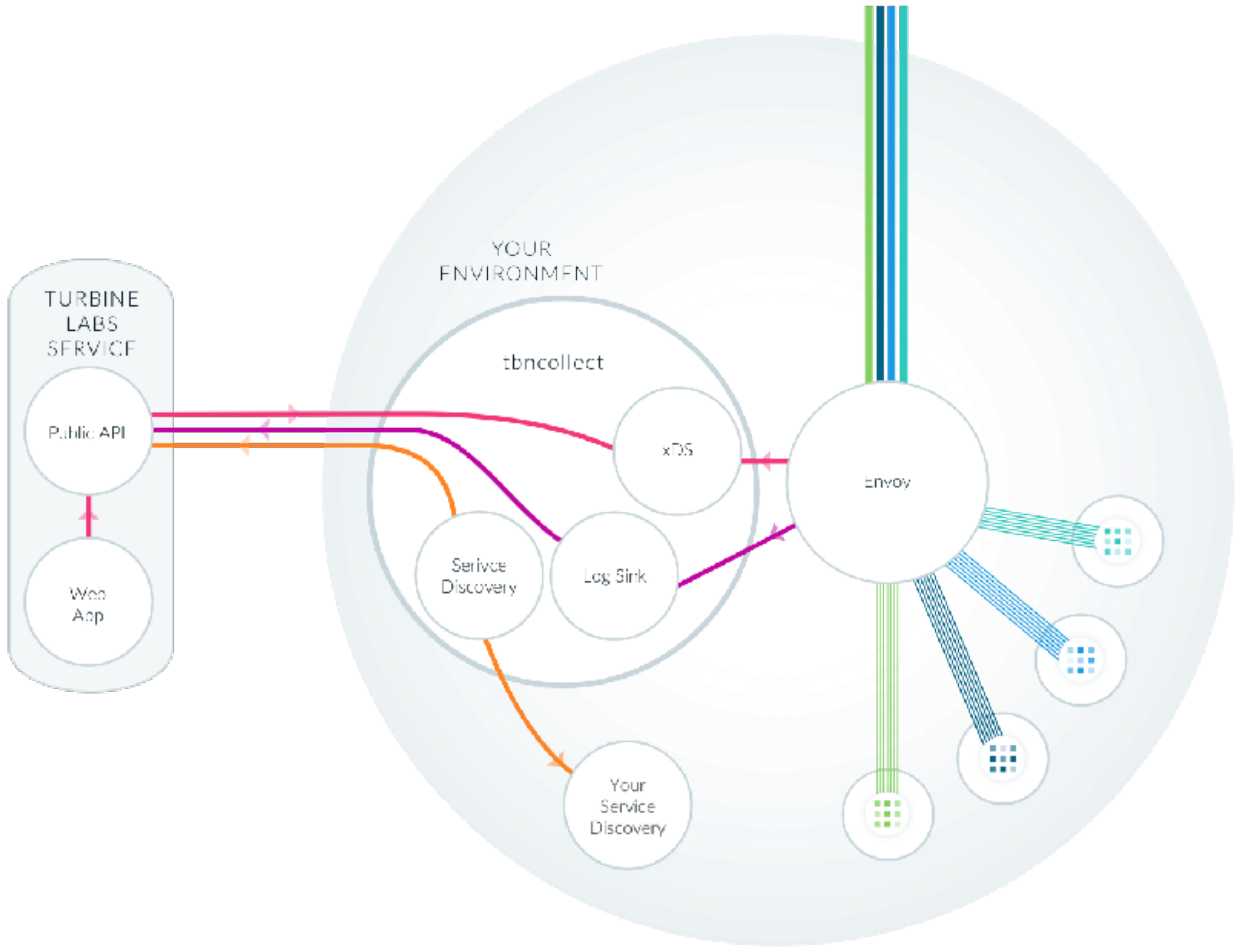

?
envoy

Envoy architecture



TURBINE LABS

Envoy benefits



- Routing primitives are more sophisticated.
- Changes are easier to get into the core project.
- Upstream definitions include arbitrary metadata.
- Config-via-service eliminates need for config management agent.
- Logs-via-gRPC eliminate need for log parsing agent.
- Wide range of protocols supported.

TURBINE LABS

We need more than a feature list

Supported

- How's the community?
- Active?
- Responsive?

Lightweight

- Are you considering sidecars?
- Memory matters!

Fast Enough

- This is important-ish
- Don't get hung up here!

Predictable

- 2 hops — 2 many?
- Performance under a variety of workloads

TURBINE LABS

Envoy has an active approachable stack community

Maintained by	lyft	Google		TURBINE LABS	
Used by	lyft	Google	IBM	verizon✓	salesforce
	ebay		Microsoft	stripe	Square
	VSCO	Tencent 腾讯	twilio	f5	airbnb
	NETFLIX	Pinterest	Medium	Booking.com	

71
committers, from 23 organizations at the last release

50%
of the commits came from the core maintaining orgs

Fast, light and predictable

C++

gives balance of speed, footprint and extensibility

Envoy has a small footprint for sidecar deployments

No garbage collector means predictable latencies

It's fast enough

Let’s do a migration!

TURBINE LABS

Planning the migration: Feature Parity

Things you should consider	How we've addressed this
NGINX and HAProxy are highly capable, highly configurable systems	

TURBINE LABS

Planning the migration: Feature Parity

Things you should consider	How we've addressed this
NGINX and HAProxy are highly capable, highly configurable systems	We started out with a higher level config abstraction to work against. Almost all constructs mapped from NGINX to Envoy nicely

Planning the migration: Feature Parity

Things you should consider	How we've addressed this
NGINX and HAProxy are highly capable, highly configurable systems	We started out with a higher level config abstraction to work against. Almost all constructs mapped from NGINX to Envoy nicely
There may be feature gaps in your move to Envoy. Fill them or avoid them: • Contribute to core Envoy • Write your own filters • Create a shared, higher level config abstraction	

TURBINE LABS

Planning the migration: Feature Parity

Things you should consider	How we've addressed this
NGINX and HAProxy are highly capable, highly configurable systems	We started out with a higher level config abstraction to work against. Almost all constructs mapped from NGINX to Envoy nicely
There may be feature gaps in your move to Envoy. Fill them or avoid them: • Contribute to core Envoy • Write your own filters • Create a shared, higher level config abstraction	Where we found gaps, we worked to fill them in Envoy itself: • Arbitrary metadata on endpoints • Subset load balancing • Mac build

TURBINE LABS

Planning the migration: Operational Parity

Logs

Metrics

Alerting

Process Management

Packaging

Deployment

TURBINE LABS

Planning the migration: “de-risk the rollout”

Sliders beat switches

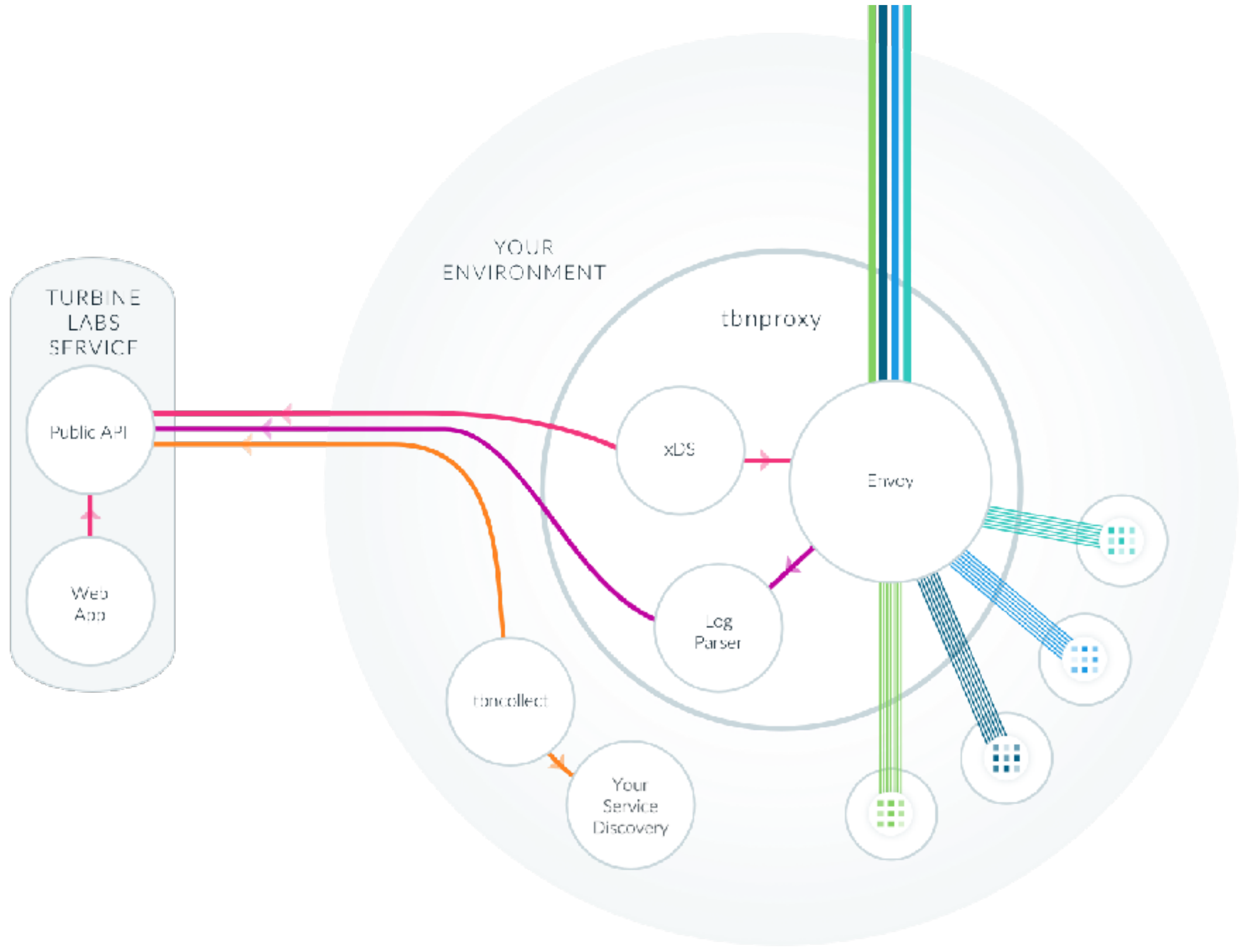
- Plan to run both systems concurrently
- Build in easy ramp up/down

Observe and compare behavior.

What observations would make us ramp down?

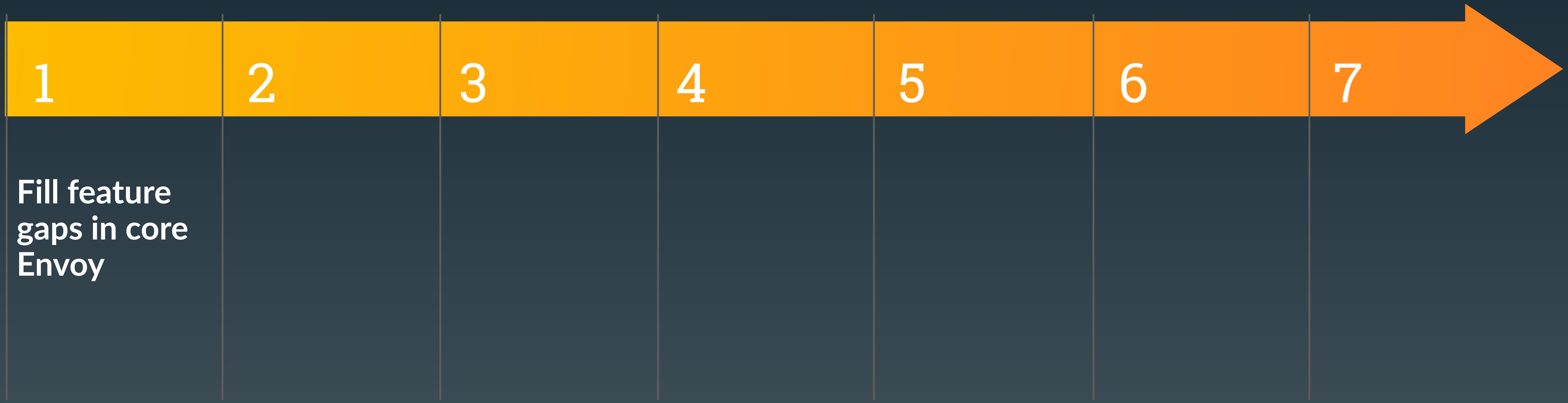
TURBINE LABS

Phase 1



- Use the same deploy ergonomics.
- Make the swap transparent
- Run xDS as a config agent
- Run a modified log parser
- This was literally just a version bump on the tbnproxy container.

Phase 1 work items



Phase 1 work items

1	2	3	4	5	6	7
Fill feature gaps in core Envoy	Implement an xDS service					

TURBINE LABS

Phase 1 work items

1	2	3	4	5	6	7
Fill feature gaps in core Envoy	Implement an xDS service	Update our log parser to work with Envoy logs.				

TURBINE LABS

Phase 1 work items

1	2	3	4	5	6	7
Fill feature gaps in core Envoy	Implement an xDS service	Update our log parser to work with Envoy logs.	Update packaging/ deployment to swap NGINX for Envoy.			

TURBINE LABS

Phase 1 work items

1	2	3	4	5	6	7
Fill feature gaps in core Envoy	Implement an xDS service	Update our log parser to work with Envoy logs.	Update packaging/ deployment to swap NGINX for Envoy.	Test. A lot.		

Phase 1 work items

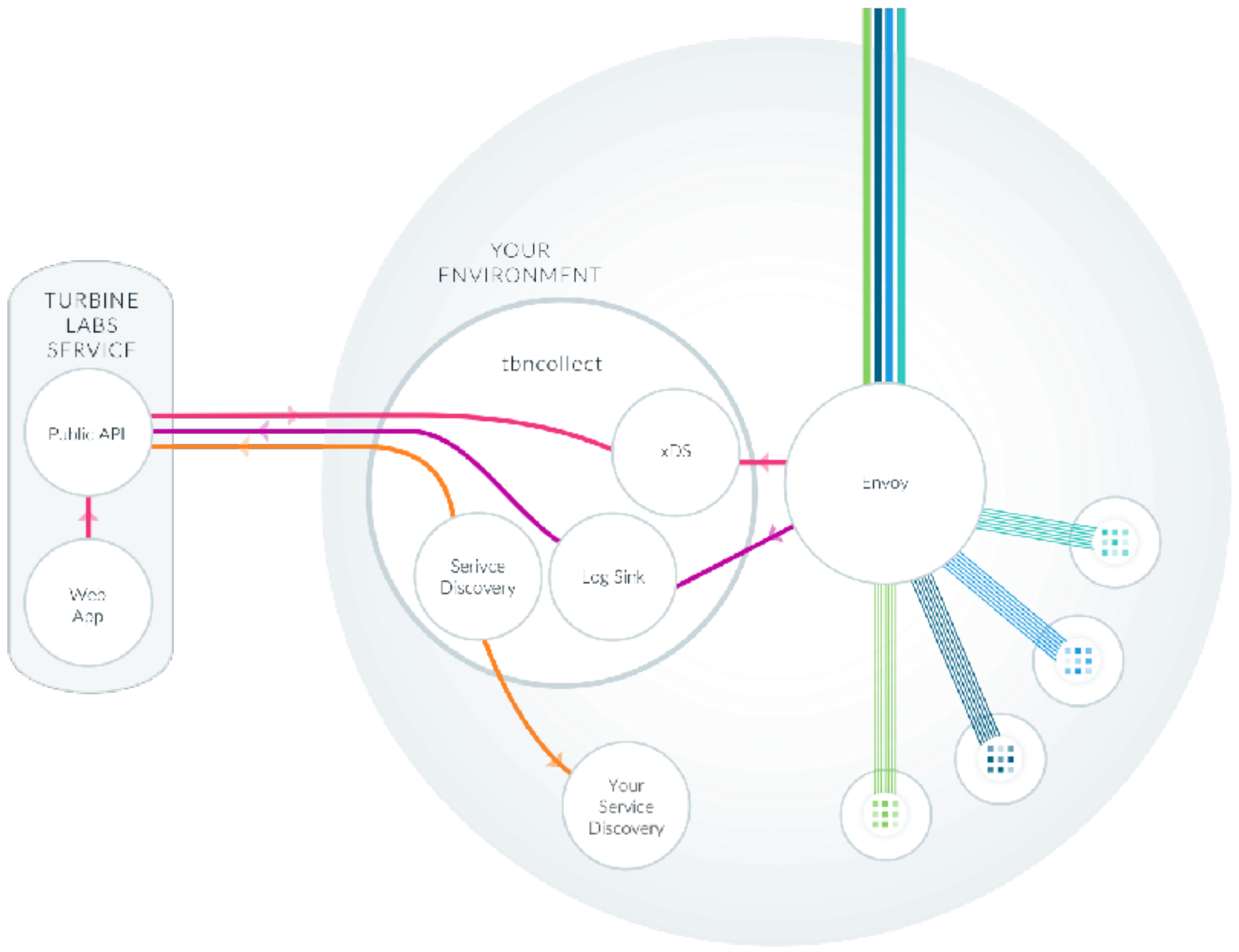
1	2	3	4	5	6	7
Fill feature gaps in core Envoy	Implement an xDS service	Update our log parser to work with Envoy logs.	Update packaging/ deployment to swap NGINX for Envoy.	Test. A lot.	Canary new proxies.	

Phase 1 work items

1. Fill feature gaps in core Envoy
2. Implement an xDS service
3. Update our log parser to work with Envoy logs.
4. Update packaging/ deployment to swap NGINX for Envoy.
5. Test. A lot.
6. Canary new proxies.
7. Fully roll out new proxies.

TURBINE LABS

Beyond parity: embracing xDS



- Once we rolled out we could take advantage of new capabilities.
- Completely separate data and control plane.
- Centralize xDS off-proxy.
- The new ALS (access log service) lets us ship telemetry over gRPC.

TURBINE LABS

After action report

TURBINE LABS

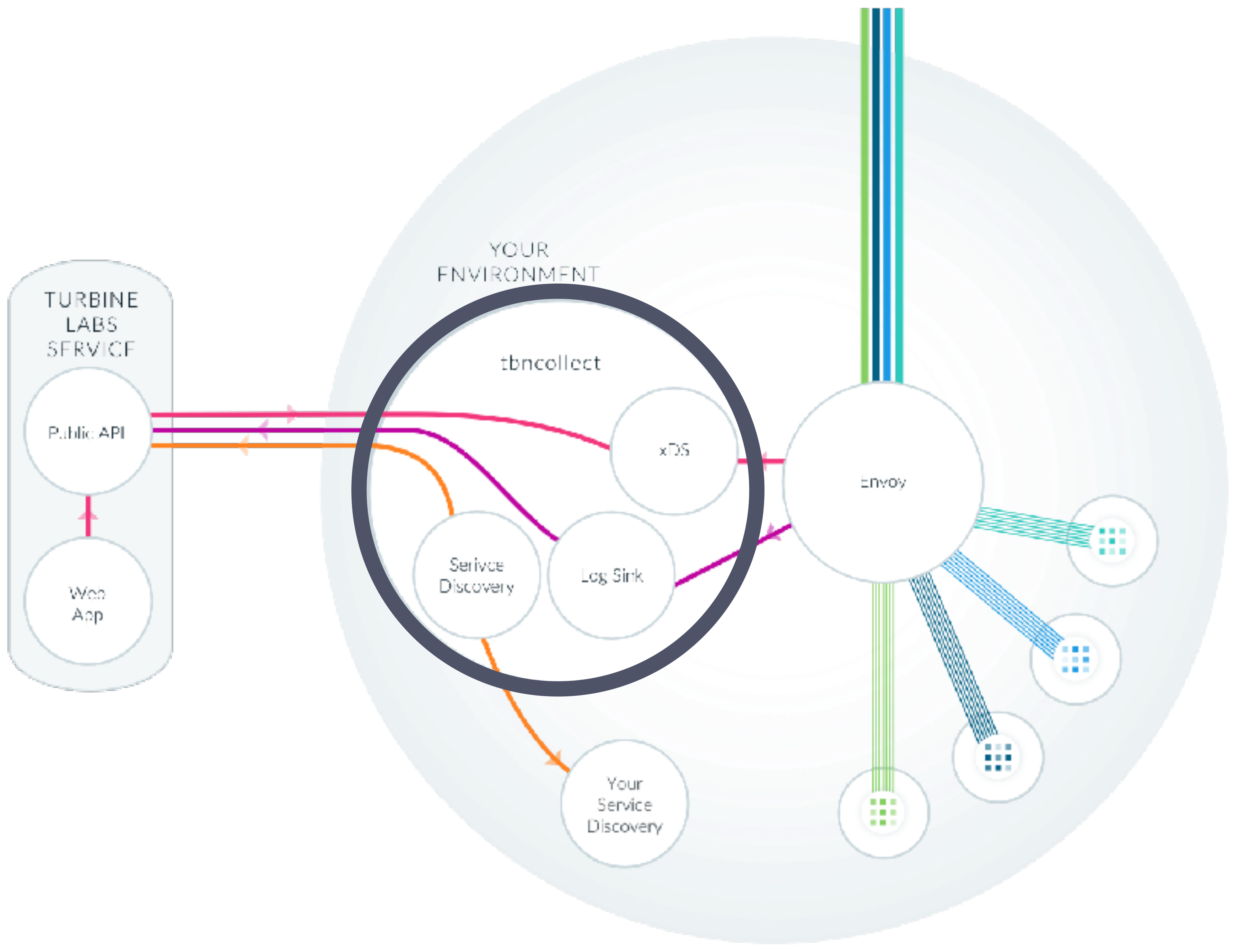
Lesson learned

- Spend more time planning than you expect to.
- Take an incremental approach.
- The difference between cluster management and listener management is subtle but important.
- Cluster management is usually a mapping from an existing source of truth to EDS/CDS.
- Listener and route management is something you own.

Not rocket science

- None of this is especially difficult.
- However a full rollout is a lengthy, high risk project.
- Stubbing your toe is easy.
- A lot of the systems we built along the way are useful outside our project.

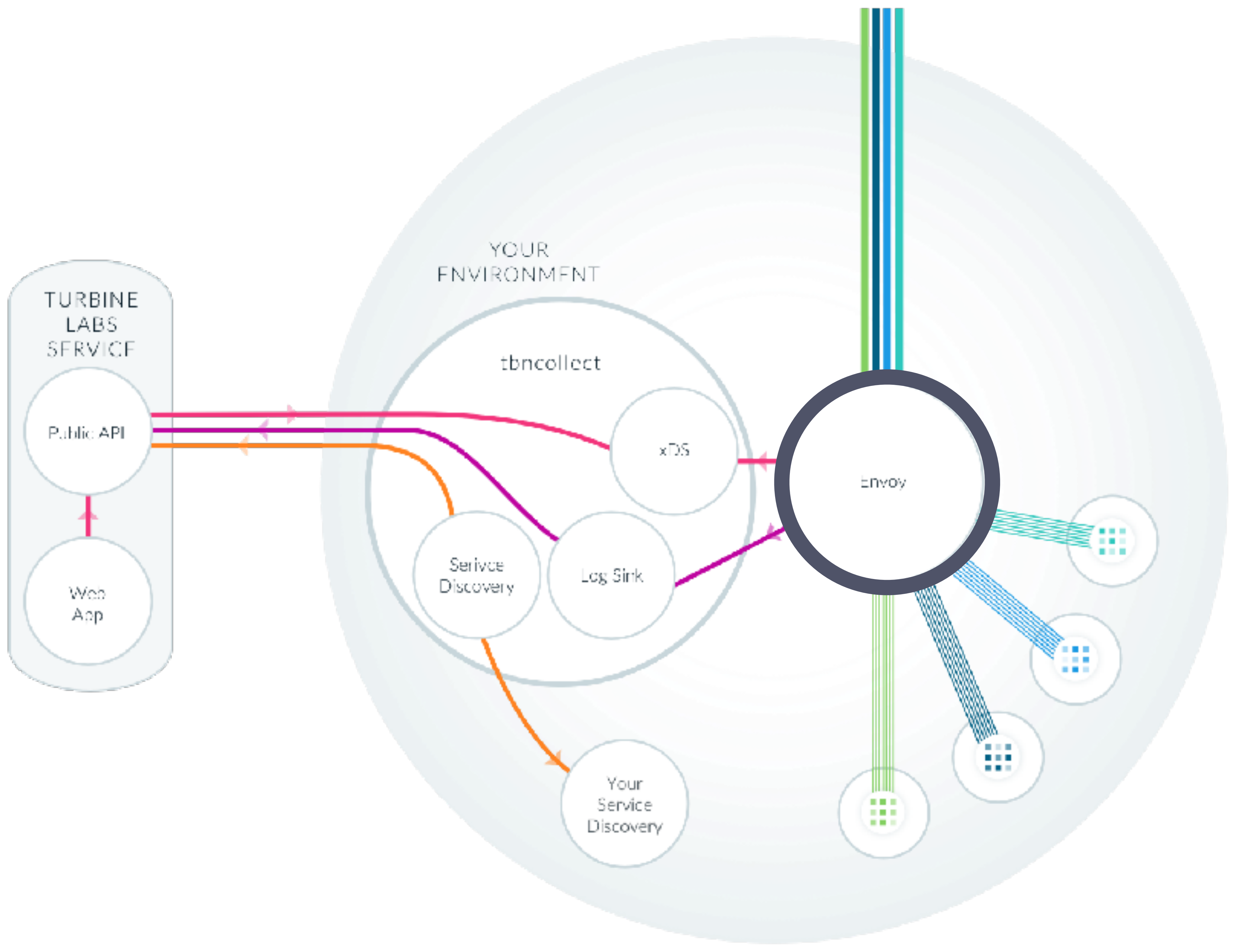
Open sourcing tbncollect



- We're open sourcing our xDS implementation, to help speed up your Envoy rollout.
- Map Kubernetes, Consul, EC2, ECS, DC/OS, and JSON/YAML files to CDS/EDS.
- ALS implementation forwards logs to statsd, Wavefront, InfluxDB or Prometheus.
- Use tbncollect's simple LDS/RDS configuration, or configure routes in a static config.
- Add a Turbine Labs product key to unlock traffic management superpowers.
- Stay informed at https://www.turbinelabs.io/tbncollect

TURBINE LABS

Open sourcing envoy-simple



- A simple container running envoy with a templated bootstrap config to connect to xDS.
- Configuration is managed via environment variables.
- Configure the Envoy process node ID, cluster, zone, and log level.
- Configure the admin server port, listener IP and log destination.
- Configure xDS ip, port, connect timeout, and refresh interval.

TURBINE LABS

Thank you

TURBINE LABS

Me:
mark@turbinelabs.io
twitter.com/mccv

Company:
turbinelabs.io
twitter.com/goturbine

envoy

Envoy project:
envoyproxy.github.io
blog.envoyproxy.io

Lessons from Envoy at scale:
learnenvoy.io

TURBINE LABS