Architecting GenServers for Testability

Tyler Young
Enbala Power Networks

Hi, I’m Tyler.

Recovering C++ developer

Fell in love with Elixir near the end of 2019

Team lead writing Elixir & Elm at Enbala (we’re hiring 😄)

“I’m not an expert, I’m just a dude.”

— Scott Schurr, CppCon 2015

You won’t agree with everything I say.

You might be right, or we might both be right, but coming from different contexts.

An exercise in comparative architecture

Separation of concerns +
Keeping closely related logic together

---

github.com/s3cur3/genserver_architecture

An exercise in comparative architecture

Reduce coupling + increase cohesion

github.com/s3cur3/genserver_architecture

An exercise in comparative architecture

Reduce coupling + increase cohesion

Explicit is better than implicit

---

github.com/s3cur3/genserver_architecture

An exercise in comparative architecture

Reduce coupling + increase cohesion

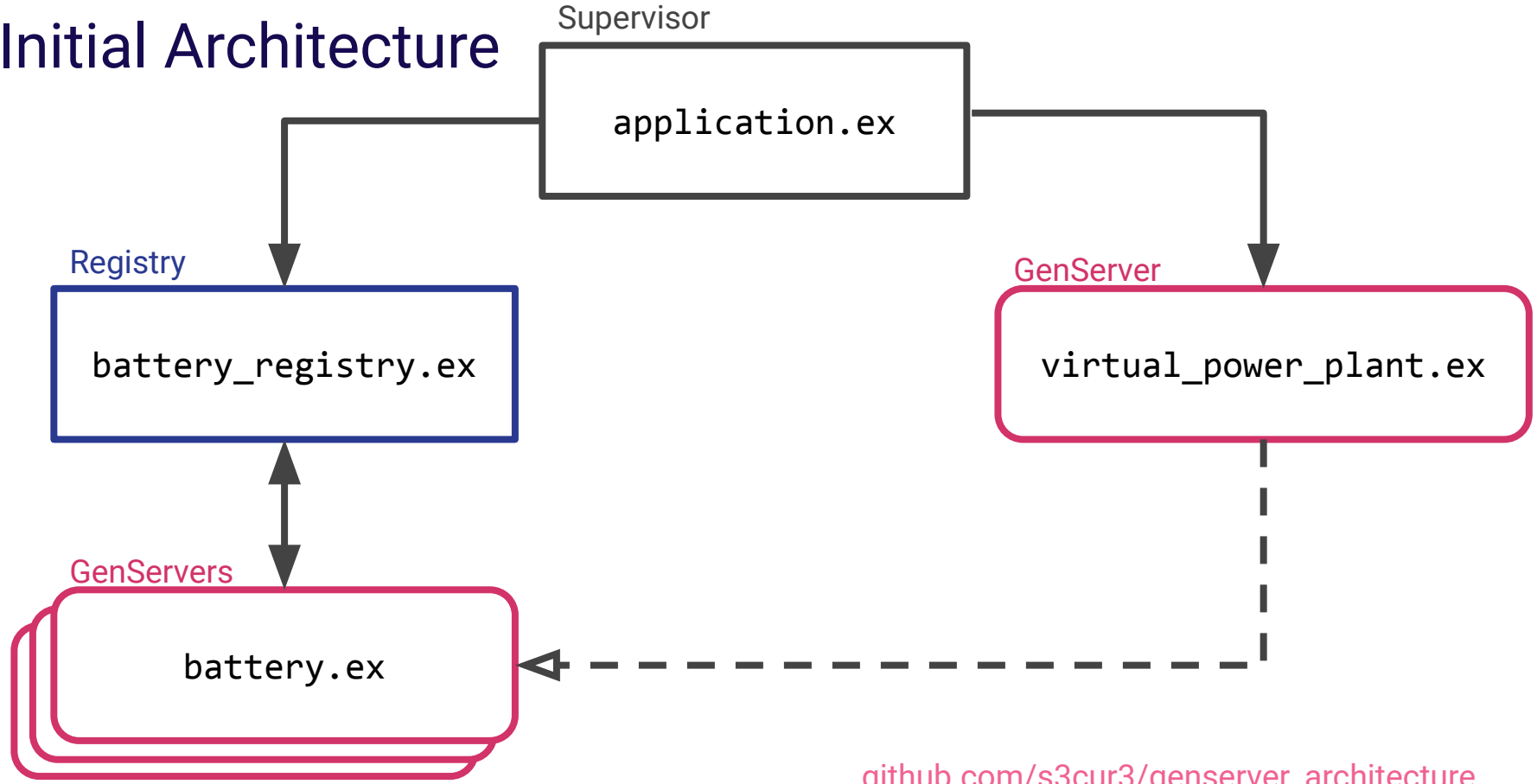
Explicit is better than implicit

Prefer `call` over `cast`

---

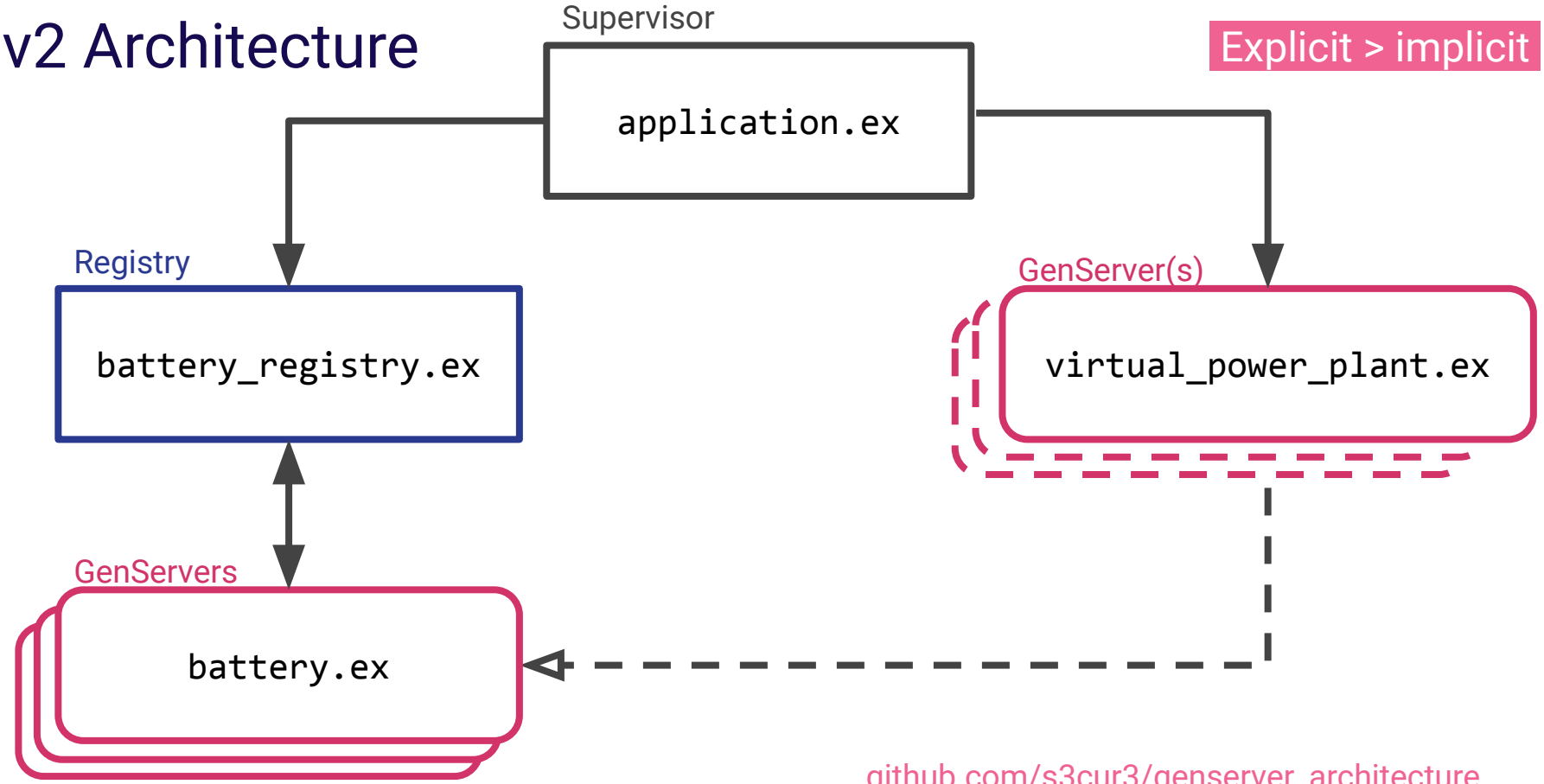
github.com/s3cur3/genserver_architecture

Initial Architecture



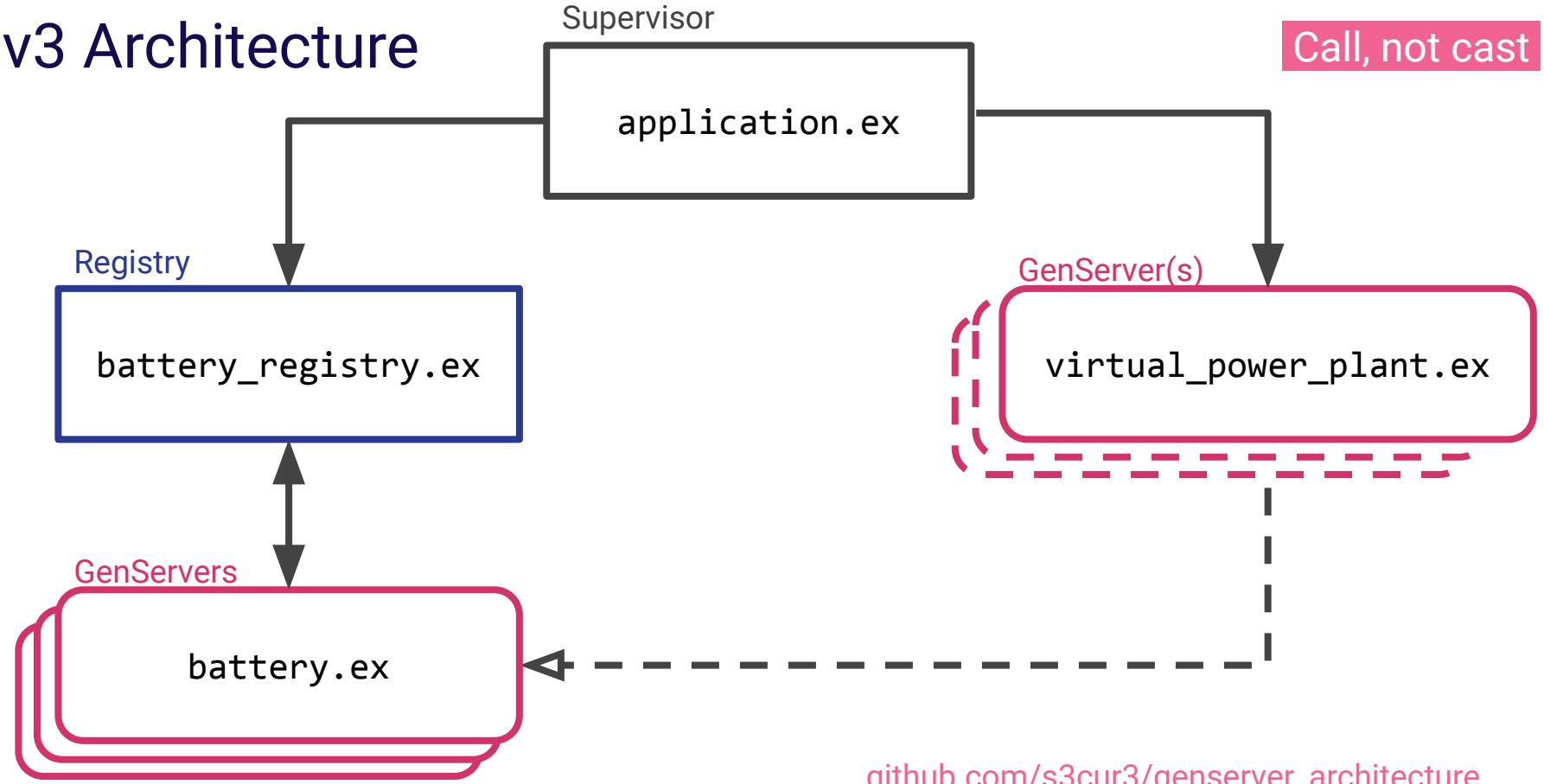
github.com/s3cur3/genserver_architecture

v2 Architecture



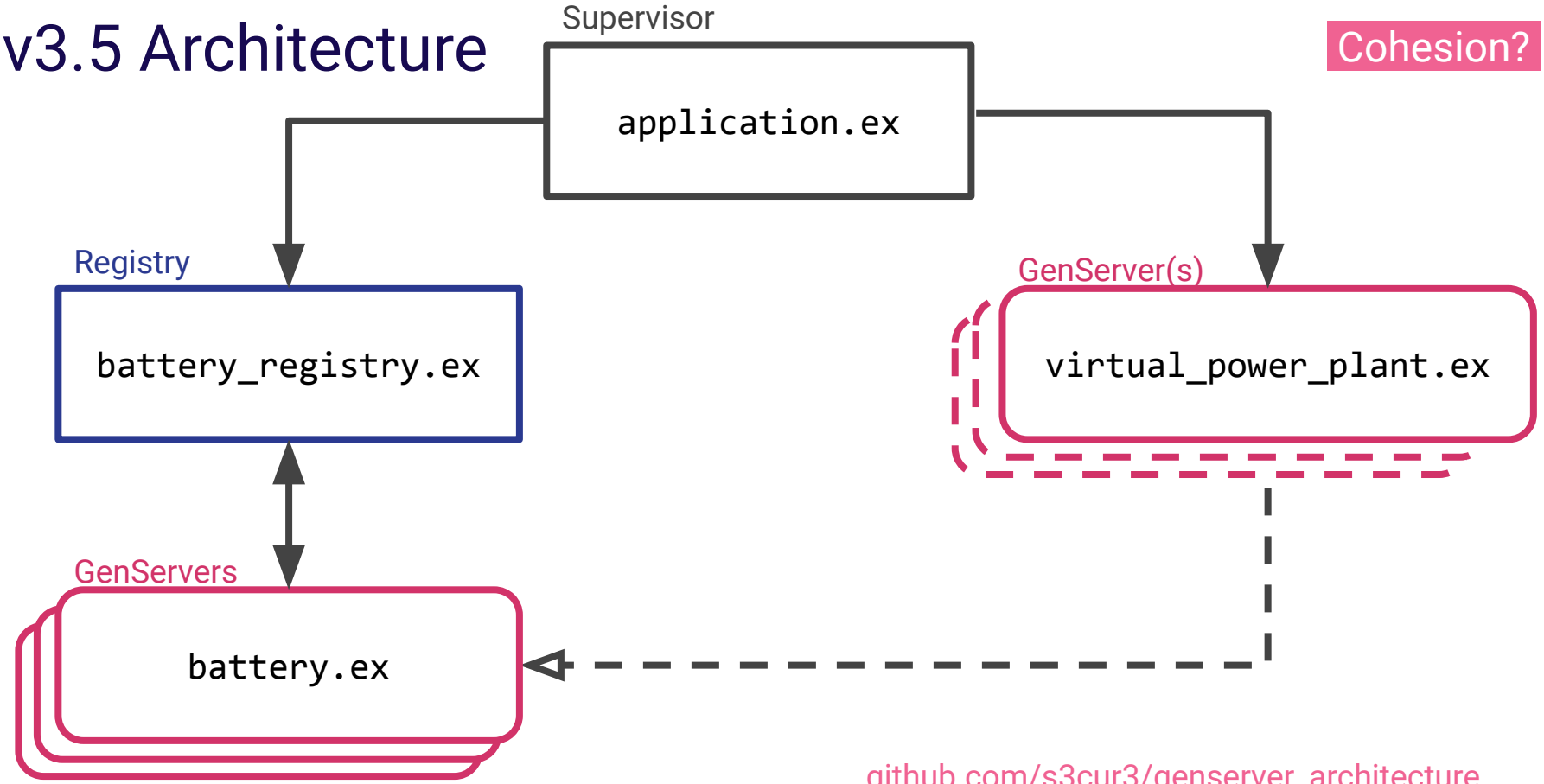
github.com/s3cur3/genserver_architecture

v3 Architecture



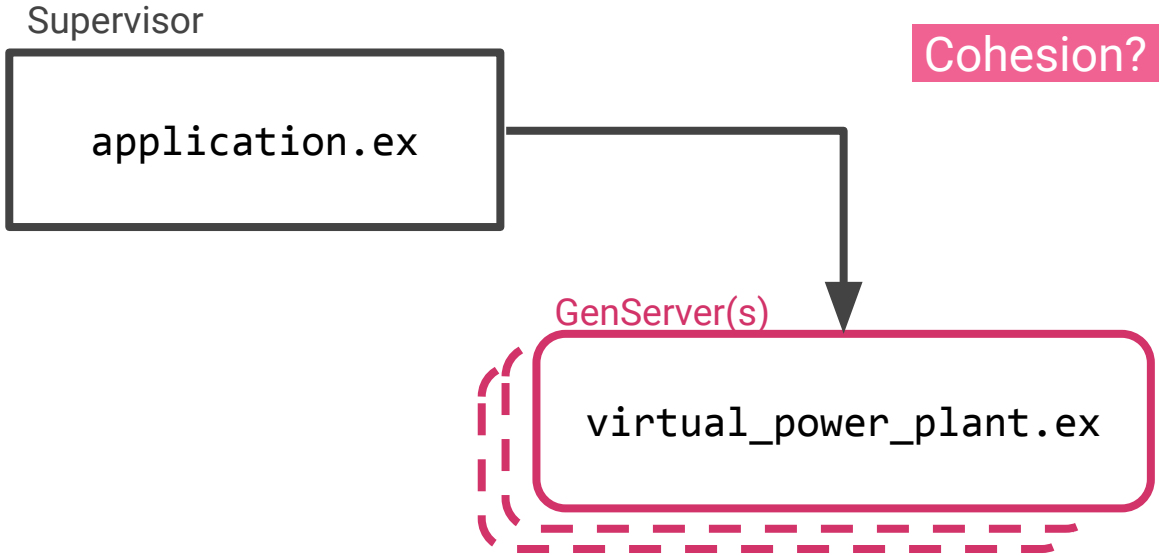
github.com/s3cur3/genserver_architecture

v3.5 Architecture



github.com/s3cur3/genserver_architecture

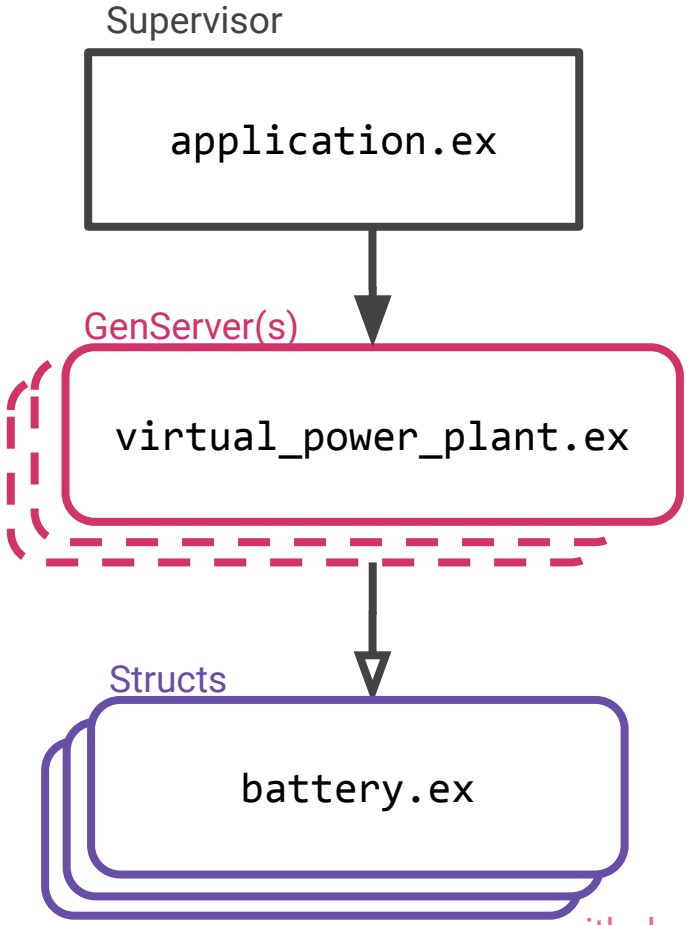
v3.5 Architecture



Cohesion?

github.com/s3cur3/genserver_architecture

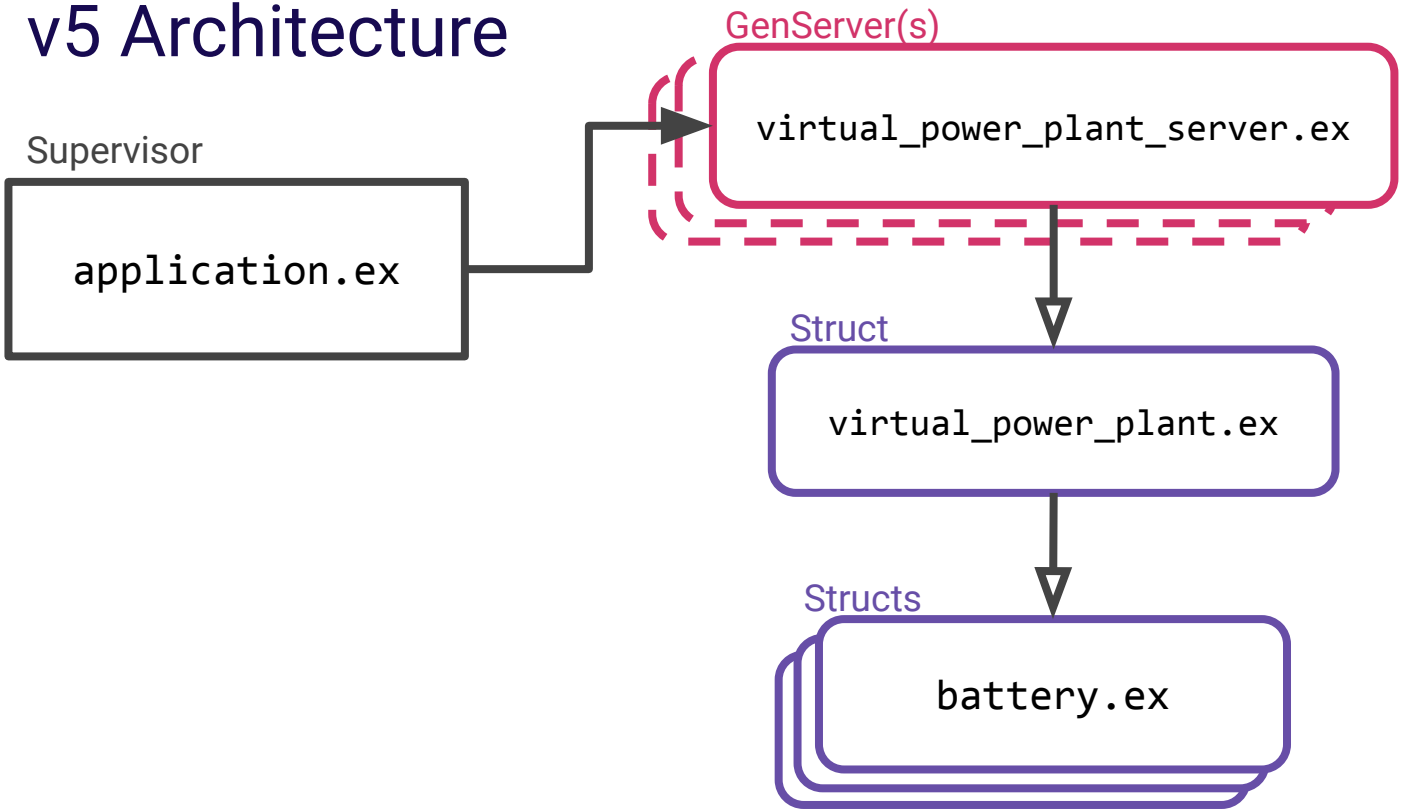
v4 Architecture



Sep. of concerns

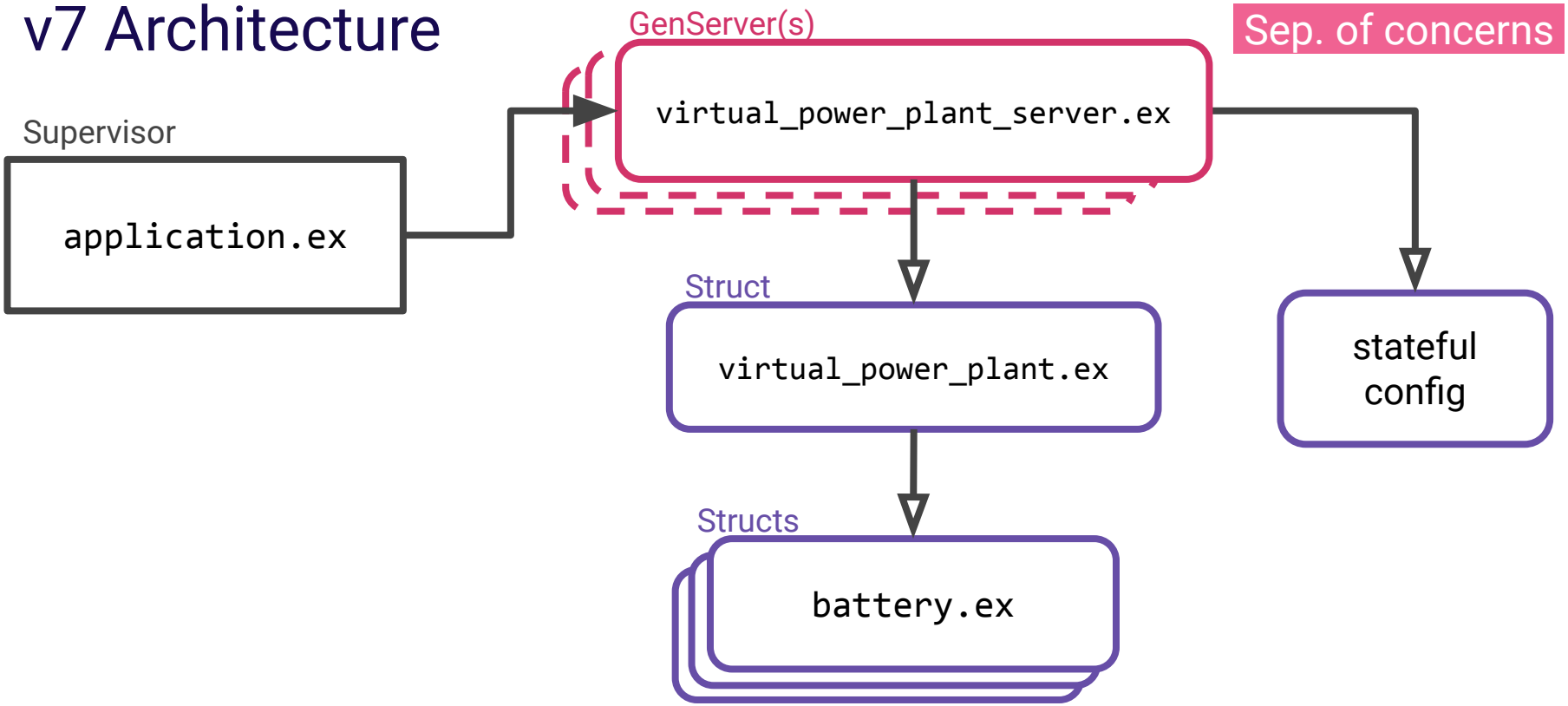
github.com/s3cur3/genserver_architecture

v5 Architecture



github.com/s3cur3/genserver_architecture

v7 Architecture



github.com/s3cur3/genserver_architecture

Recap

- Reduce coupling + increase cohesion
- Prefer `call` over `cast`
- Explicit is better than implicit

🐦 @TylerAYoung on Twitter with questions & comments

💻 Enbala is always hiring Elixir devs!
⚡ enbala.com/careers/