STPA at Meta

Todd Porter (Meta)

Meta

Outline

This talk presents, in detail, the STPA journey that Meta has been on over the last year:

- Initial interest, attending a STAMP workshop, online training
- Revisiting two significant software weaknesses to ask the question “could STPA have helped to catch this?”
- Applying STPA to a recent design
- Getting buy off and hosting an initial STPA training class
- Learnings from the training class and next steps

Meta

Training

- Attended workshop at MIT
- Took pilot online training course
- Worked directly with Professor John Thomas (MIT) to review two recent incidents and a new design

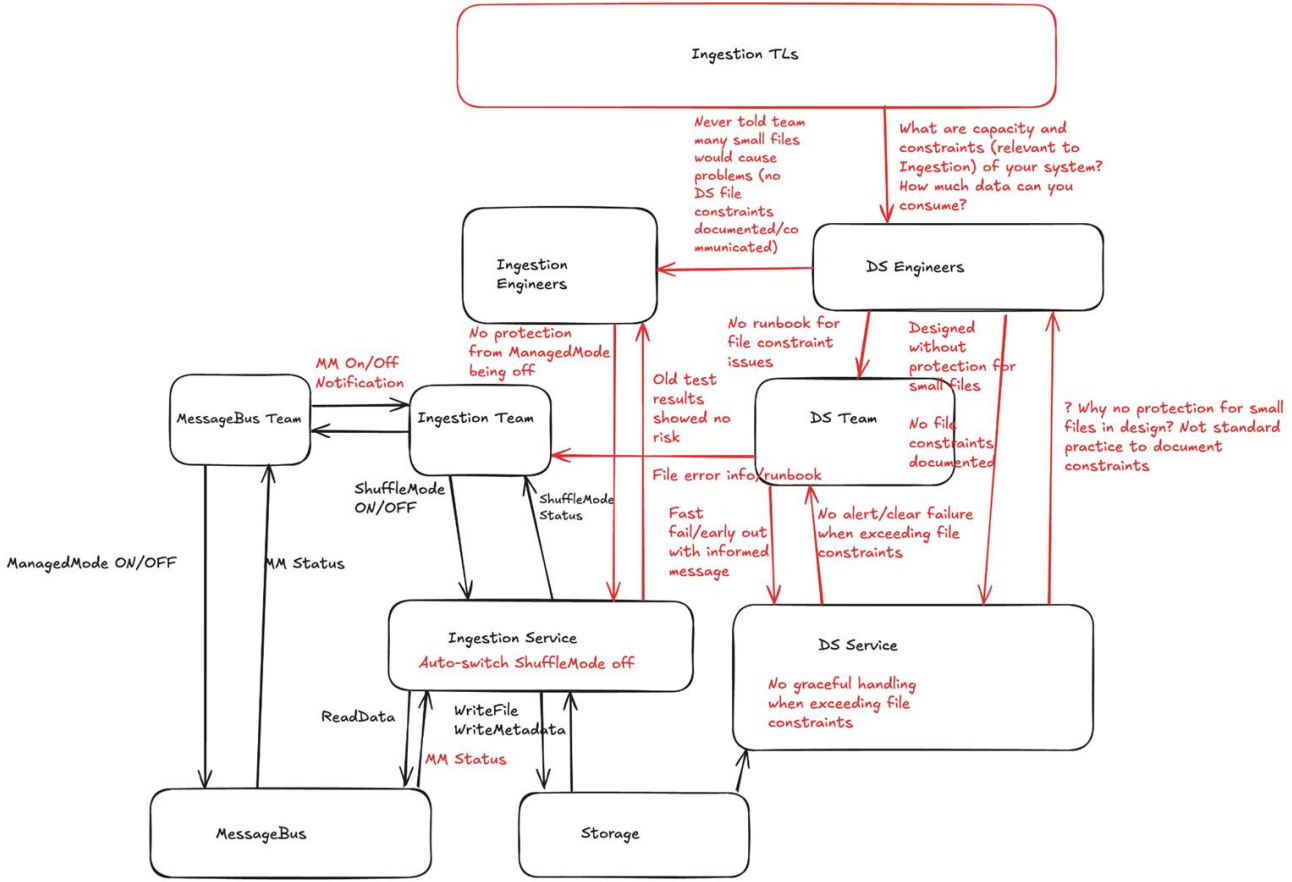
Meta

Two recent incidents, could STPA have helped?

- Analyzed a pair of recent incidents where our data ingestion platform was creating too many small files for downstream systems

Meta

Two recent incidents, could STPA have helped? Yes!

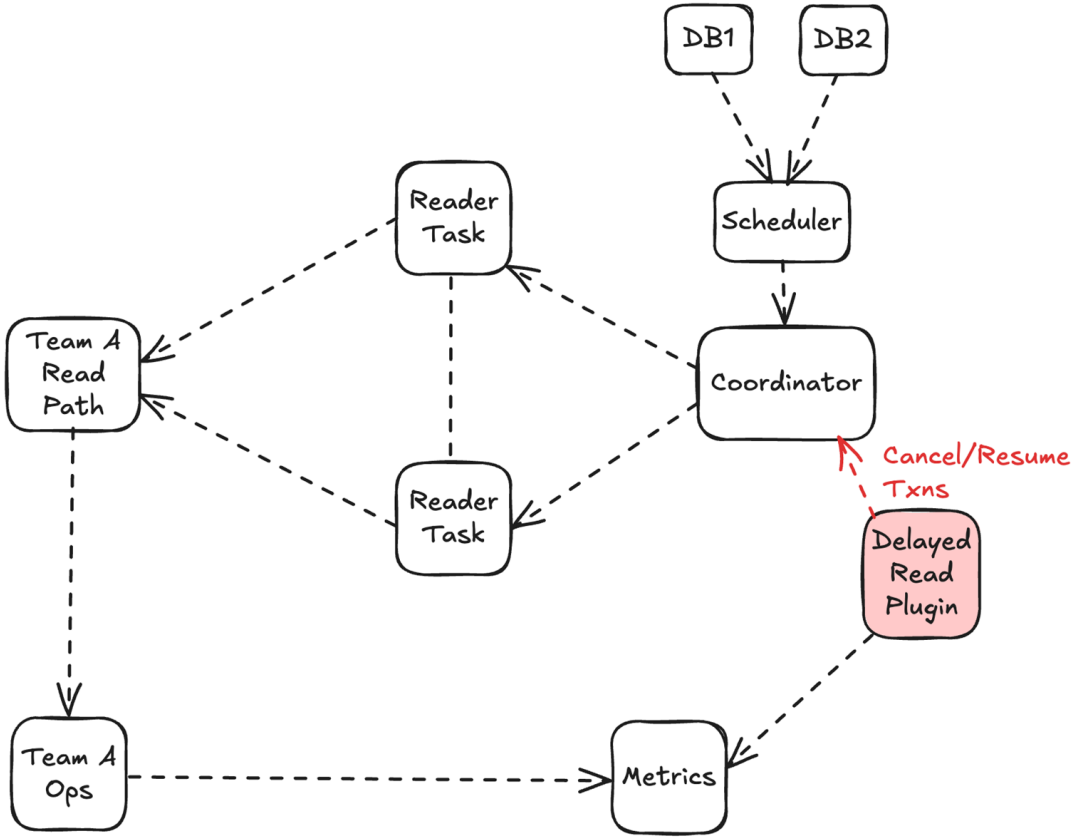


New design, can STPA provide value?

- STPA provided value for previous incidents, but what about in the design phase?
- Applied it to a new feature, Delayed Reads, which was still in the design phase

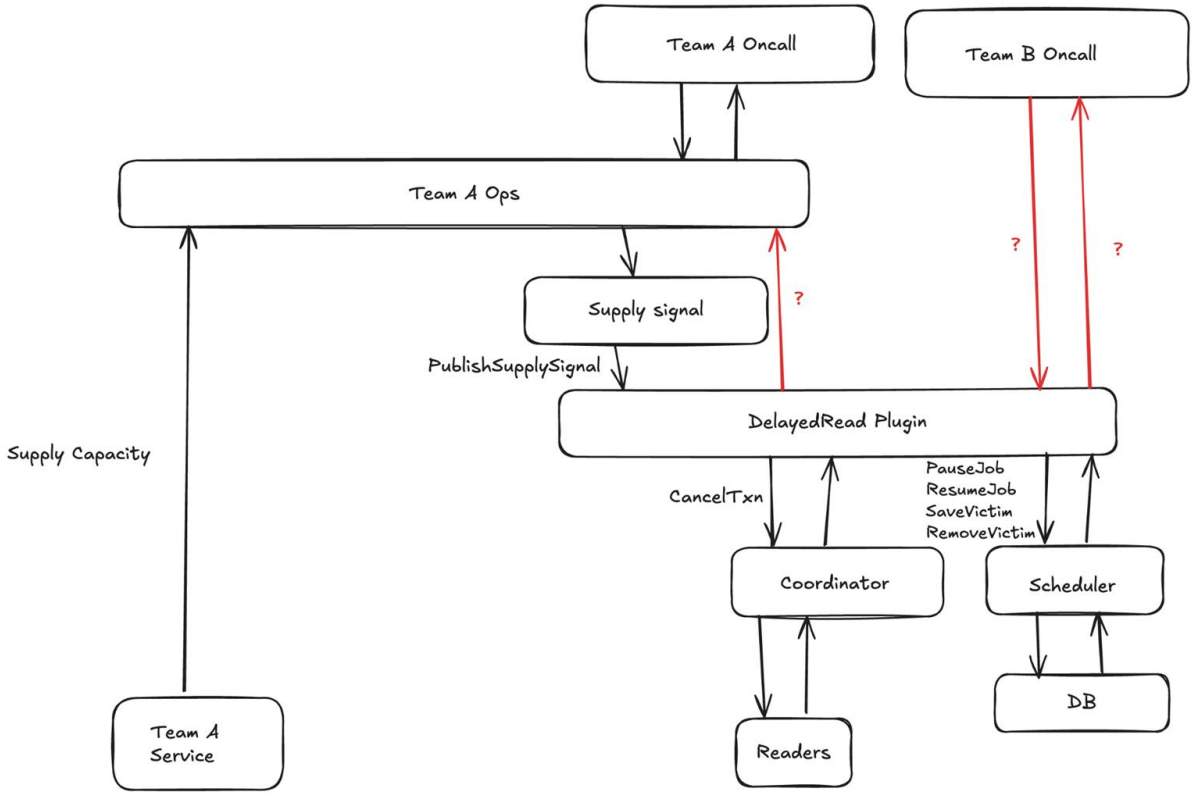
Meta

New design, can STPA provide value?



Meta

New design, can STPA provide value? Yes!



New design, can STPA provide value?

- Took one UCA that was immediately identified and looked for it in the code, found the exact section of code to look at, confirmed code was actually always susceptible to the UCA

Meta

Initial training class

- Results from quick STPA analysis:
  - Informative and actionable results were found on recent incidents and a new design, this piqued a lot of interest in STPA
  - The core concept of a control structure, modeling a system with control-feedback loops, was a very natural abstraction that resonated deeply
- Management and TLs were intrigued and wanted to continue with STPA
- We had our first 12 person STPA training class in December 2025

Meta

Learnings and next steps

- Engineers liked the experience overall
  - Would prefer more tailored examples
- Teams have already applied it to ~5 projects and found value in every one
  - The feedback path always could use more rigor
- Many canonical control loop scenarios
  - Quota enforcement
  - Autoscaling
  - Overload protection
- Having scenario based STPA examples could greatly accelerate learning and adoption

Meta

Questions?