

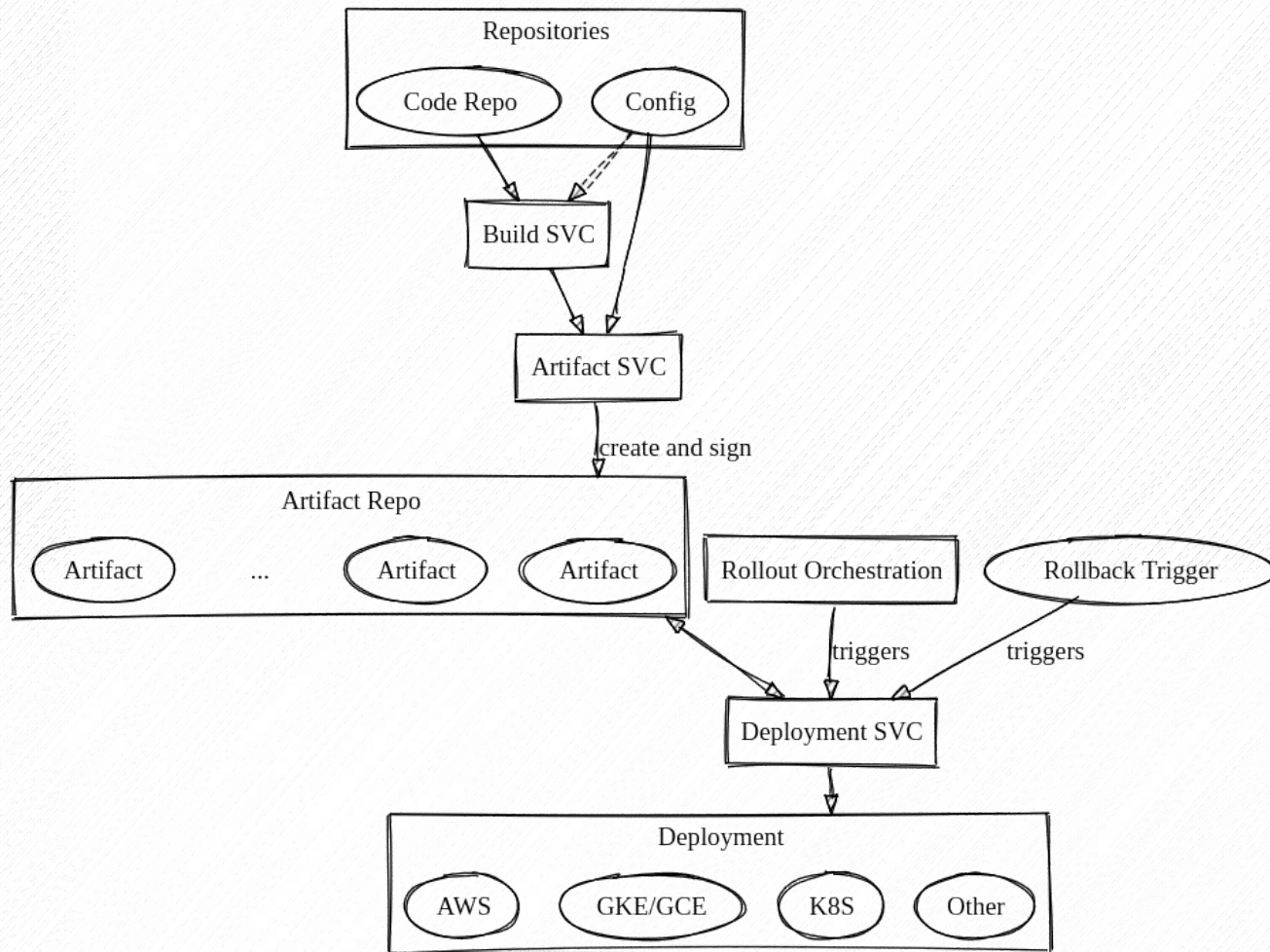
Production Undo

Why (fast) Rollbacks?

- Fast rollbacks allow you to stay calm when rolling out
- Easiest way to mitigate an active outage
- Forces you to rethink your architecture and the way you deploy software
- Faster rollbacks allow for faster rollouts

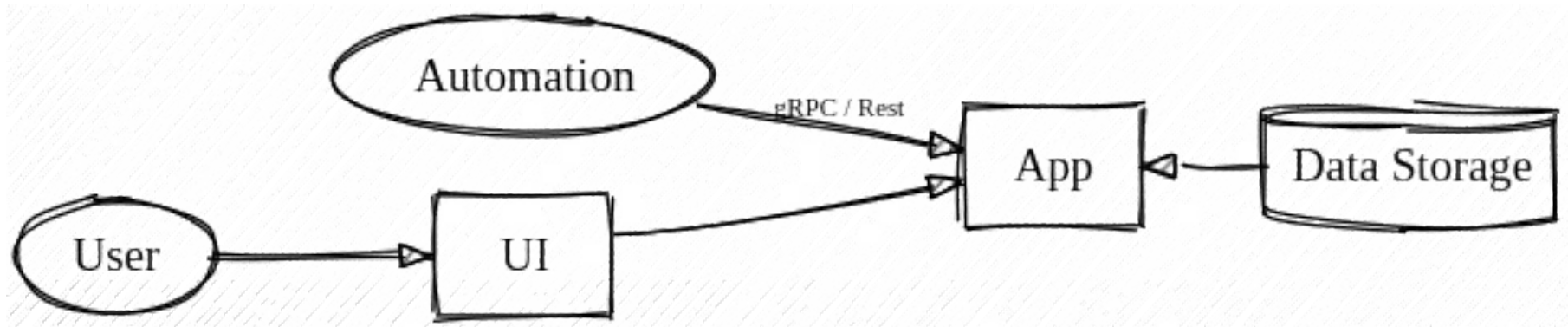
Assumptions

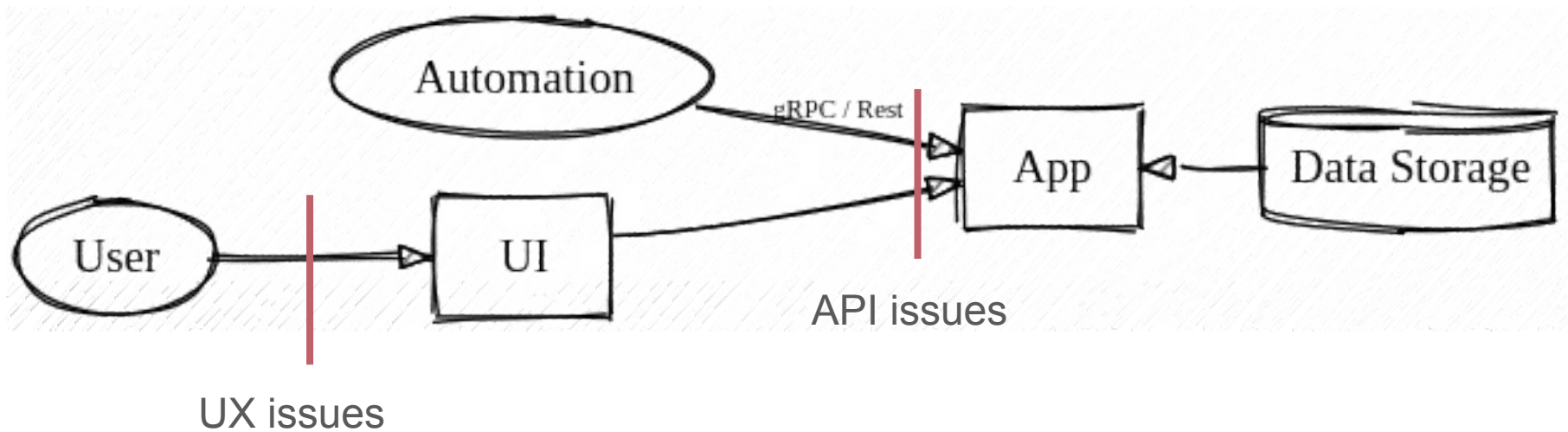
1. We have rollout automation in place
2. We have an SLO to defend
3. We monitor said SLO



A few ground rules

1. Code and Config are part of the artifact
2. Artifacts are either stored immutable or can easily be reproduced
3. Each artifact can be rolled out independently
4. Rollout orchestration should (ideally) be per artifact
5. Your rollout process must keep track of what version is currently being deployed
6. You must have immutable storage of the artifact version history (we need to know which version was last known good)



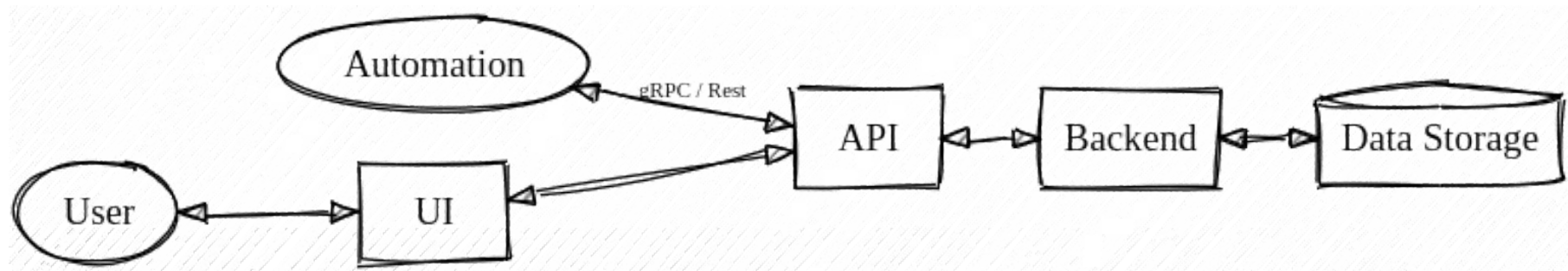


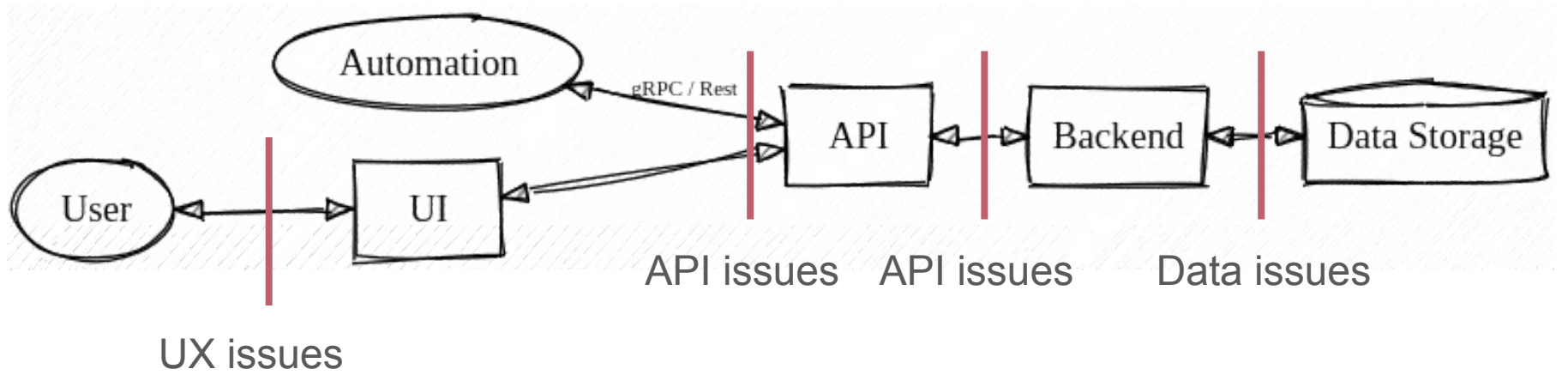
API Issues and how to avoid them

- Assume APIs are external
- Version your APIs
- Forwards and Backwards compatibility is crucial
 - Avoid changing the parameters in your API calls
 - Preserve but ignore fields that your code does not understand
- Don't reinvent the wheel, use existing concepts

UX Issues

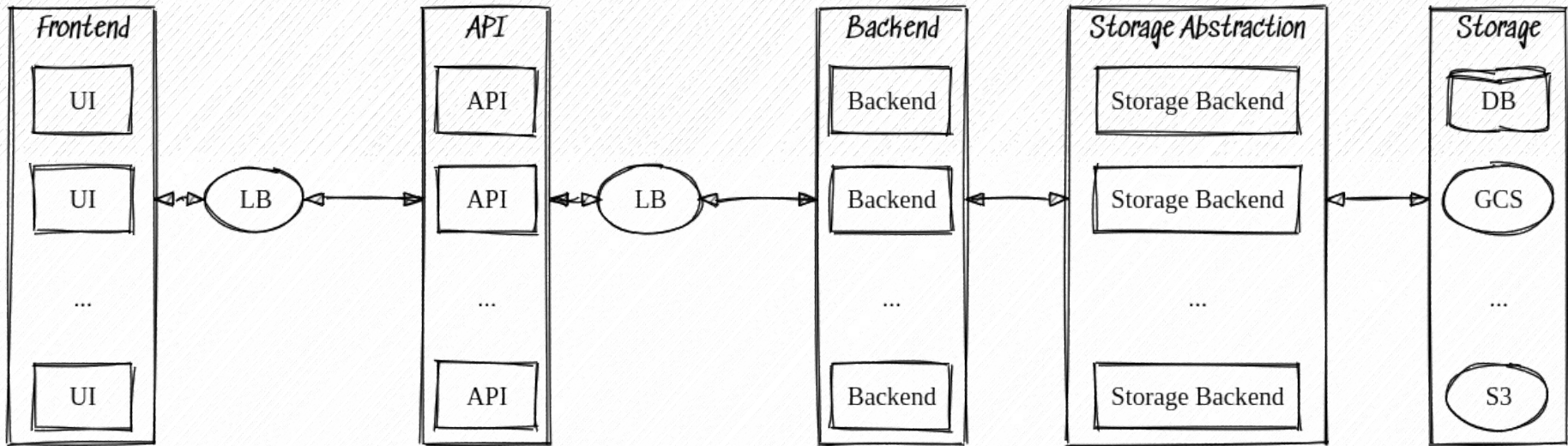
- Simply roll back

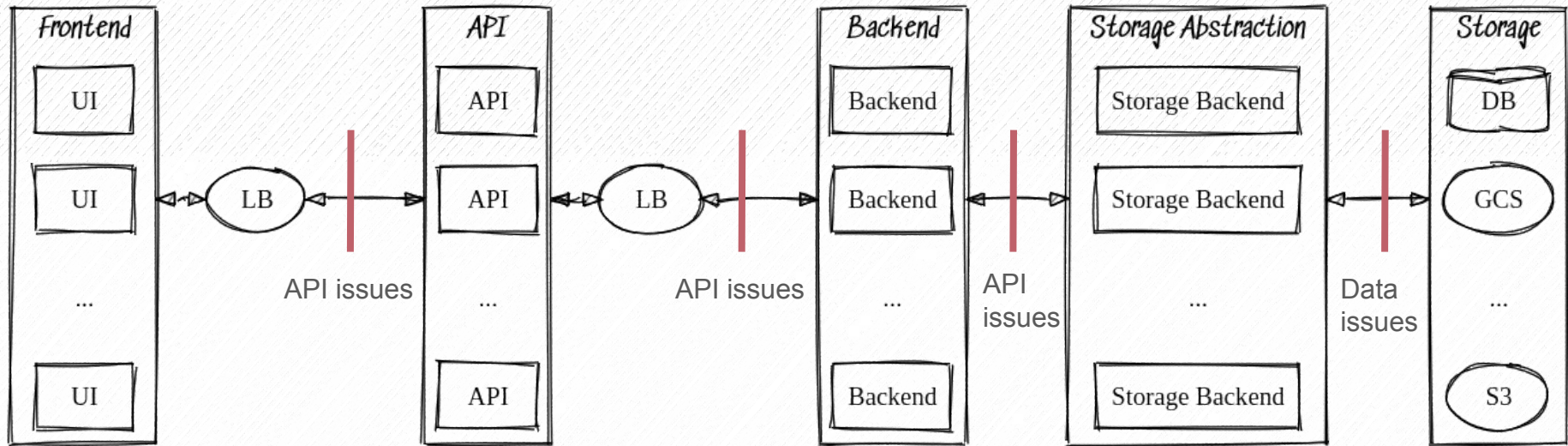




A word on Storage

- You (probably) do not need a database
 - If the data is mostly read and comfortably fits into memory, think about distributing it as a file
 - If it is config data it should be a file (or multiple)
- Generally avoid having multiple writers (as in code locations not concurrency)
- Abstract away your storage layer
 - Easier to update
 - For-/Backwards compatibility is easier to implement
- If you need to use a DB, think about your schema **a lot**





Takeaways

- Make sure you understand what can be deployed independently
- UX issues can be solved by blindly rolling back
- API issues can be prevented by having versioned APIs and fixed by being forwards and backwards compatible
- Data issues can be mitigated by
 - Being forwards and backwards compatible
 - Having a backup from before the rollout and an append-only log
- Test your rollbacks regularly
- Have backups and test your restore procedures

Questions?