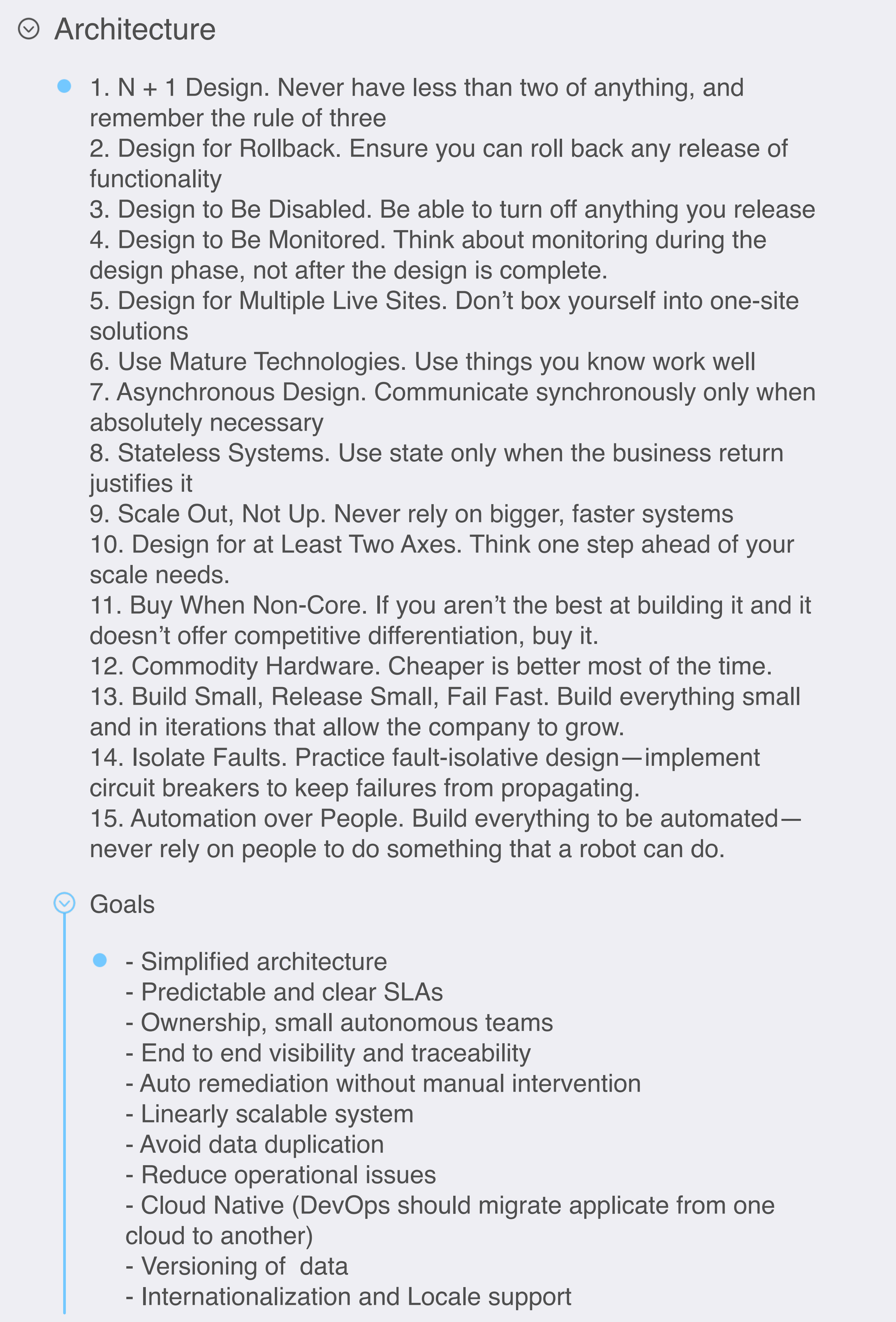

Architecture

1. N + 1 Design. Never have less than two of anything, and remember the rule of three
2. Design for Rollback. Ensure you can roll back any release of functionality
3. Design to Be Disabled. Be able to turn off anything you release
4. Design to Be Monitored. Think about monitoring during the design phase, not after the design is complete.
5. Design for Multiple Live Sites. Don’t box yourself into one-site solutions
6. Use Mature Technologies. Use things you know work well
7. Asynchronous Design. Communicate synchronously only when absolutely necessary
8. Stateless Systems. Use state only when the business return justifies it
9. Scale Out, Not Up. Never rely on bigger, faster systems
10. Design for at Least Two Axes. Think one step ahead of your scale needs.
11. Buy When Non-Core. If you aren’t the best at building it and it doesn’t offer competitive differentiation, buy it.
12. Commodity Hardware. Cheaper is better most of the time.
13. Build Small, Release Small, Fail Fast. Build everything small and in iterations that allow the company to grow.
14. Isolate Faults. Practice fault-isolative design—implement circuit breakers to keep failures from propagating.
15. Automation over People. Build everything to be automated—never rely on people to do something that a robot can do.

Goals

- Simplified architecture
- Predictable and clear SLAs
- Ownership, small autonomous teams
- End to end visibility and traceability
- Auto remediation without manual intervention
- Linearly scalable system
- Avoid data duplication
- Reduce operational issues
- Cloud Native (DevOps should migrate applicate from one cloud to another)
- Versioning of data
- Internationalization and Locale support