*For context on this tool, read* [Decision Log @ Infrastructure Engineering](https://infraeng.dev/decision-log/)

Infra Decision Log

*Questions? Reach out in #decisions-infra on Slack*

This document tracks open and finalized decisions for the Infrastructure organization.

# Open Decisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Question** | **DRI** | **Date Added** | **Notes** |
| Our migration away from the monolith is in year two and almost all new feature work still requires monolith changes. Should we revisit our strategy? | Michelle | 2022/1/10 | Discuss in #decision-monolith |
| How should we prioritize migrating frontend Javascript to TypeScript? | Elizabeth | 2022/1/17 | Discuss in #decision-typescript |
| Should we migrate our internal event collector and storage to Segment? | Jeremy | 2022/2/1 | Discuss in #decision-segment |

# Finalized Decisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Decision** | **DRI** | **Date Decided** | **Notes** |
| Approved backend languages are Go, Python | Michelle | 2021/12/5 | Backend Language Working Group Summary |
| Ongoing education budget is $1500/person/year | Jeremy | 2021/2/11 | Ongoing Education Policy Update |
| SREs share career ladder with Infrastructure Engineers | Elizabeth | 2021/1/5 | Why We’re Sharing Career Ladder Between SRE and Infra Eng |

**Interpretations**

**Technical Spike**

A spike in a sprint can be used in a number of ways:

* As a way to familiarize the team with new hardware or software
* To analyze a problem thoroughly and assist in properly dividing work among separate team members.
* Spike tests can also be used to mitigate future risk, and may uncover additional issues that have escaped notice.

A distinction can be made between technical spikes and functional spikes. The technical spike is used more often for evaluating the impact new technology has on the current implementation. A functional spike is used to determine the interaction with a new feature or implementation.

[Engineering feasibility spikes](https://playbook.microsoft.com/code-with-engineering/design/design-reviews/recipes/engineering-feasibility-spikes/) can also be conducted to de-risk an engagement and increase the team's understanding.

**Exception constructs**[¶](https://playbook.microsoft.com/code-with-engineering/design/exception-handling/readme/%2523exception-constructs)

Almost all language platforms offer a construct of exception or equivalent to handle error scenarios. The underlying platform, used libraries or the authored code can "throw" exceptions to initiate an error flow. Some of the advantages of using exceptions are -

1. Abstract different kind of errors
2. Breaks the control flow from different code structures
3. Navigate the call stack till the right catch block is identified
4. Automatic collection of call stack
5. Define different error handling flows thru multiple catch blocks
6. Define finally block to cleanup resources