Вплоть до момента начала работы над модулем метрик весь код был написан на JavaScript (TypeScript). Решение о его использовании было принято из-за предыдущего опыта, большого сообщества и простоты в использовании. Его известные недостатки были несущественны для остальных модулей.

Постановка задач для модуля метрик заставила задуматься о том, стоит ли и этот модуль писать на JS, потому что проблемы языка для этого модуля гораздо более принципиальны.

**Речь идёт только о модуле вычисления метрик.** Сама подсистема для метрик будет включать в себя сервер для общения административной панели и модуля вычисления метрик. Этот сервер будет написан на JS. Он не будет требовать большого объёма вычислительных ресурсов, поэтому я не вижу смысла полностью переходить на новый язык, на котором у меня нет богатого опыта разработки.

При этом библиотеку (фреймворк) для сервера на JS я решил заменить с Express (используется для бизнес-логики в основной части сервиса) на NestJS. В ретроспективе мне показалось использование Express ошибкой, пусть и не столь принципиальной. Библиотека Express рассчитана скорее на небольшие сервера с парой эндпоинтов и не имеет строгих архитектурных ограничений и правил. В течение разработки эти правила пришлось искусственно создавать, на выходе получилась архитектура, слабо отличающаяся от NestJS. С учётом того, что в NestJS эти правила встроены и хорошо задокументированы, для масштабирования этот фреймворк подойдёт лучше.

Возвращаясь к модулю вычисления метрик, JS очевидно не подходит. Нужен более быстрый язык, способный параллельно выполнять задачи. В JS есть асинхронность, но она отвечает только за то, в какой момент выполнение одной части программы заменяется на выполнение другой. Несколько процедур одновременно JS выполнять не может.

| Требование\Язык | JS | php | Go | python | Rust | Java |
| --- | --- | --- | --- | --- | --- | --- |
| Наличие опыта использования | + | + | - | +- | - | - |
| Простота в изучении | + | +- | + | + | - | - |
| Скорость | - | +- | + | - | + | +- |
| Возможность использования HTTP-запросов | + | +- | + | + | +- | + |
| Возможность исполнения shell-команд | +- | + | + | + | +- | +- |
| Многопоточность | - | +- | + | +- | + | + |
| Простота использования многопоточности | - | - | + | +- | - | +- |
| Документация | + | - | +- | - | + | +- |

Оценки всех пунктов были нормализованы относительно кандидатов (на основе субъективного удобства/простоты использования или производительности), например, Java сама по себе не слишком сложна в изучении, но в сравнении с JS, go, python и php она сложнее.