1 Auswahl der gewählten Frameworks

Der Techstack des DMF's besteht aus einer Treesitter, Golang und LSP4ij. Im weiteren werden Gründe für die Auswahl der verschieden Framworks/Tools aufgezählt.

1.1 Golang

Golang ist eine modernen Sprache deren Schwerpunkt auf die Entwicklung von Webservern und einfacher Parallelisierung liegt. Hierbei ist Golang's besonders große und leistungsstarke Standardbibliothek bemerkenswert. So werden für die meisten Aufgaben eines Webservers, keine zusätzlichen Abhängigkeiten benötigt.
Golang besitzt zudem die Möglichkeit sehr einfach C-Bindings einzubauen.
Für die Entwicklung des DMF's waren die einfache Einbindung von Treesitter, die einfache und nachvollziehbare Implementierung des JSON-RPC Protokolls und die schnelle Generierung von Dateien mithilfe von Go-Templates die ausschlaggebenden Gründe. Die Möglichkeit den LSP-Server auch über Internet-Protokolle erreichbar zu machen, kommt zudem hinzu.

1.2 Treesitter

Treesitter ist ein Framwork zu Generation von Parsern. Dafür wird mithilfe der Javascript-API eine Grammatik definiert. Aus dieser wird ein Parser in C generiert. Dieser Parser kann nun über Binding's in verschiedene Sprachen eingebunden werden.
Das Framework bietet viele Grammatiken an. Zudem gibt es eine moderne Dokumentation und Foren zu direkten Kommunikation.
Treesitter unterscheidet sich von anderen Parsern wie ANTLR durch die Möglichkeit des Iterativen Parsings. Dies ermöglicht deutlich schnelleres Parsing bei kleinen Änderungen in einer Datei, wie es häufig bei Editieren der Fall ist.
Ein Nachteil von Treesitter ist, dass der Generierte Parser in C Generiert wird und somit keinen Garbage Collector besitzt. Verbindet man nun diesen Code mit Sprache wie Golang, welche einen Garbage Collector besitzt und darauf aufgelegt ist, können Situationen entstehen, wo Memory Leaks entstehen und schwer zu debuggen sind.

1.3 LSP4ij

LSP4ij nimmt beim Namen starke Inspiration von LSP4J einer Bibliothek zur Entwicklung von LSP Clients und Servern in Java. Doch anstatt einer generelle API, handelt es sich bei LSP4IJ um ein Intelij Plugin welches die Einbindung von LSP's deutlich vereinfacht. Es ist zusätzlich möglich auf LSP4ij aufbauend ein Plugin für Intellij zu entwickeln.

Das Intellij nach VSCode die wichtigste IDE ist, hat der Support auf der Plattform hohe Priorität.