

# Chapter 1

## Table of contents

- 1 Introduction
  - 1.1 Motivation
    - \* 1.1.1 Problem Definition
  - 1.2 Research Questions
  - 1.3 Non-Goals
  - 1.4 Research Methodology
  - 1.5 Structure of the thesis
- 2 Background
  - 2.1 Language Server Protocol
    - \* 2.1.1 JSON-RPC
    - \* 2.1.2 Commands and Notifications
    - \* 2.1.3 Description of Key Methods
      - 2.1.3.1 Code Completion
      - 2.1.3.2 Hover
      - 2.1.3.3 Jump to Definition
      - 2.1.3.4 Find References
      - 2.1.3.5 Workspace/Document symbols
      - 2.1.3.6 Diagnostics
    - \* 2.1.4 File Processing
  - 2.2 Text based Configuration
    - \* 2.2.1 Common Configuration Languages
      - 2.2.1.1 JSON
      - 2.2.1.2 YAML
    - \* 2.2.2 Configuration *Programming* Languages
    - \* 2.2.3 Infrastructure as Code
    - \* 2.2.4 Nickel
      - 2.2.4.1 Nickel AST
      - 2.2.4.2 Record Merging
      - 2.2.4.3 Gradual typing
      - 2.2.4.4 Contracts
- 3 Related work
  - 3.1 IDE Support
    - \* 3.1.1 Native and Plugin Systems

- \* 3.1.2 Server Client Abstractions
      - 3.1.2.1 Monto
      - 3.1.2.2 Merlin
  - 3.2 Language Servers
    - \* 3.2.1 Integrating with the Compiler/Runtime
      - 3.2.1.1 HLS
      - 3.2.1.2 Ocaml LSP
      - 3.2.1.3 Rust-Analyzer
      - 3.2.1.4 Frege LSP
      - 3.2.1.5 Runtime-independent LSP implementations
      - 3.2.1.6 Language Server as an Interface to CLI tools
      - 3.2.1.7 CPAchecker
      - 3.2.1.8 CodeCompass
    - \* 3.2.2 Language Servers generation for Domain Specific Languages
  - 3.3 Alternative approaches
    - \* 3.3.1 LSP Extensions
    - \* 3.3.2 Language Server Index Format
    - \* 3.3.3 \*SP, Abstracting software development processes
  - 3.4 References
- 4 Design and Implementation
  - 4.1 Key Objectives
    - \* 4.1.1 Performance
    - \* 4.1.2 Capability
    - \* 4.1.3 Flexibility
    - \* 4.1.4 Generalizability
  - 4.2 Design Decisions
    - \* 4.2.1 Programming language
    - \* 4.2.2 File processing
    - \* 4.2.3 Code Analysis
  - 4.3 High-Level Architecture
  - 4.4 Illustrative example
  - 4.5 Linearization
    - \* 4.5.1 States
    - \* 4.5.2 Transfer from AST
      - 4.5.2.1 Usage Graph
      - 4.5.2.2 Scopes
      - 4.5.2.3 Linearizer
      - 4.5.2.4 Linearization Process
    - \* 4.5.3 Post-Processing
      - 4.5.3.1 Sorting
      - 4.5.3.2 Resolving deferred access
      - 4.5.3.3 Resolving types
    - \* 4.5.4 Resolving Elements
      - 4.5.4.1 Resolving by position
      - 4.5.4.2 Resolving by ID
      - 4.5.4.3 Resolving by scope
  - 4.6 LSP Server Implementation
    - \* 4.6.1 Server Interface
    - \* 4.6.2 Diagnostics and Caching
    - \* 4.6.3 Commands

- 4.6.3.1 Hover
  - 4.6.3.2 Jump to Definition and Show references
  - 4.6.3.3 Completion
  - 4.6.3.4 Document Symbols
- 5 Evaluation
  - 5.1 Methods
    - \* 5.1.1 Qualitative
    - \* 5.1.2 Quantitative
  - 5.2 Results
    - \* 5.2.1 Process
    - \* 5.2.2 Qualitative
      - 5.2.2.1 Pre-Evaluation
      - 5.2.2.2 Experience Survey
    - \* 5.2.3 Quantitative
      - 5.2.3.1 Dataset
      - 5.2.3.2 Big Picture Latencies
      - 5.2.3.3 Special cases
- 6 Discussion
  - 6.1 Project results
  - 6.2 Project shortcomings
  - Future Work

