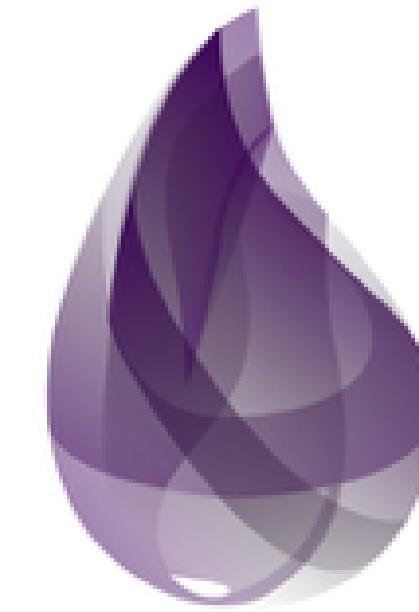


Elixir

Presented by @Dawei Ma



elixir

My programming language learning path

- C/C++
- Java/Scala (SpringBoot/Play2/Vert.x/RxJava)
- Swift/Objective-C (RxSwift)
- Python (Django/Flask/Tornado)
- Elixir/Erlang (Phoenix)
- Golang
- Javascript (Vue/Angular)
- PHP (Laravel)
- CommonLisp (Learning)
- Rust (Will Learn)

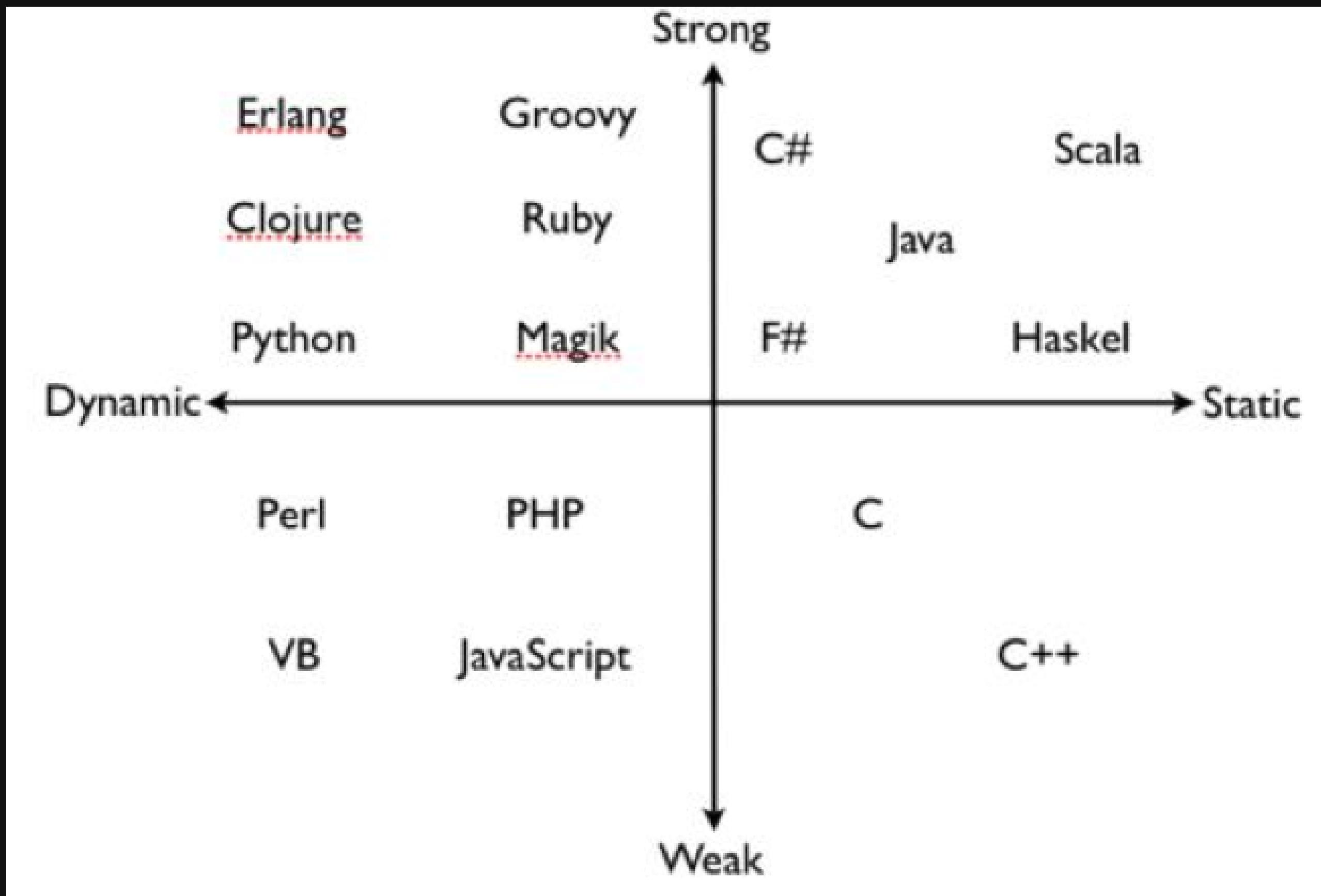
Rules for learning a language

- Must solve a problem
- Problem not addressed by current stack
- “Fast” is not a purpose
- Should be the best solution

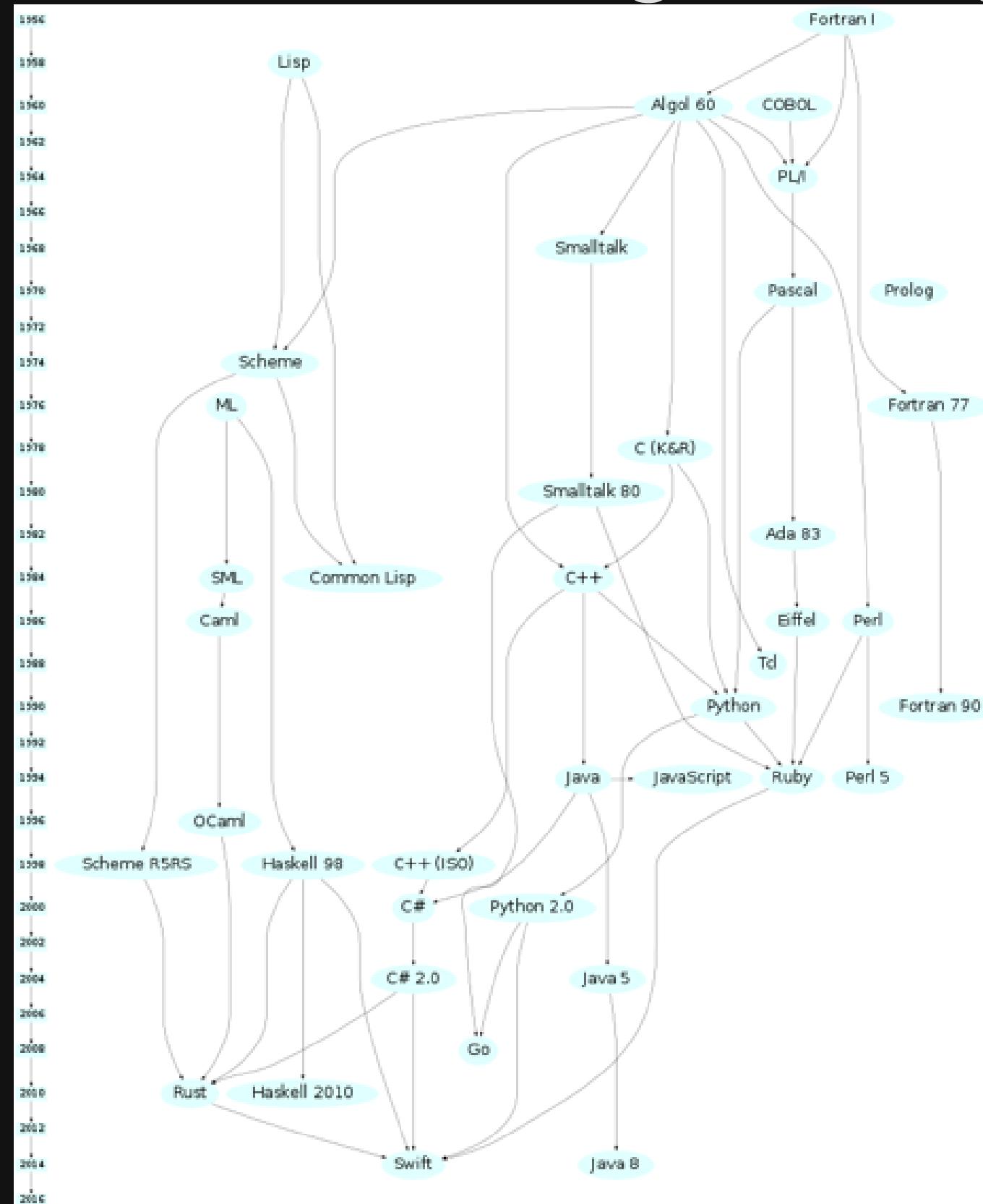




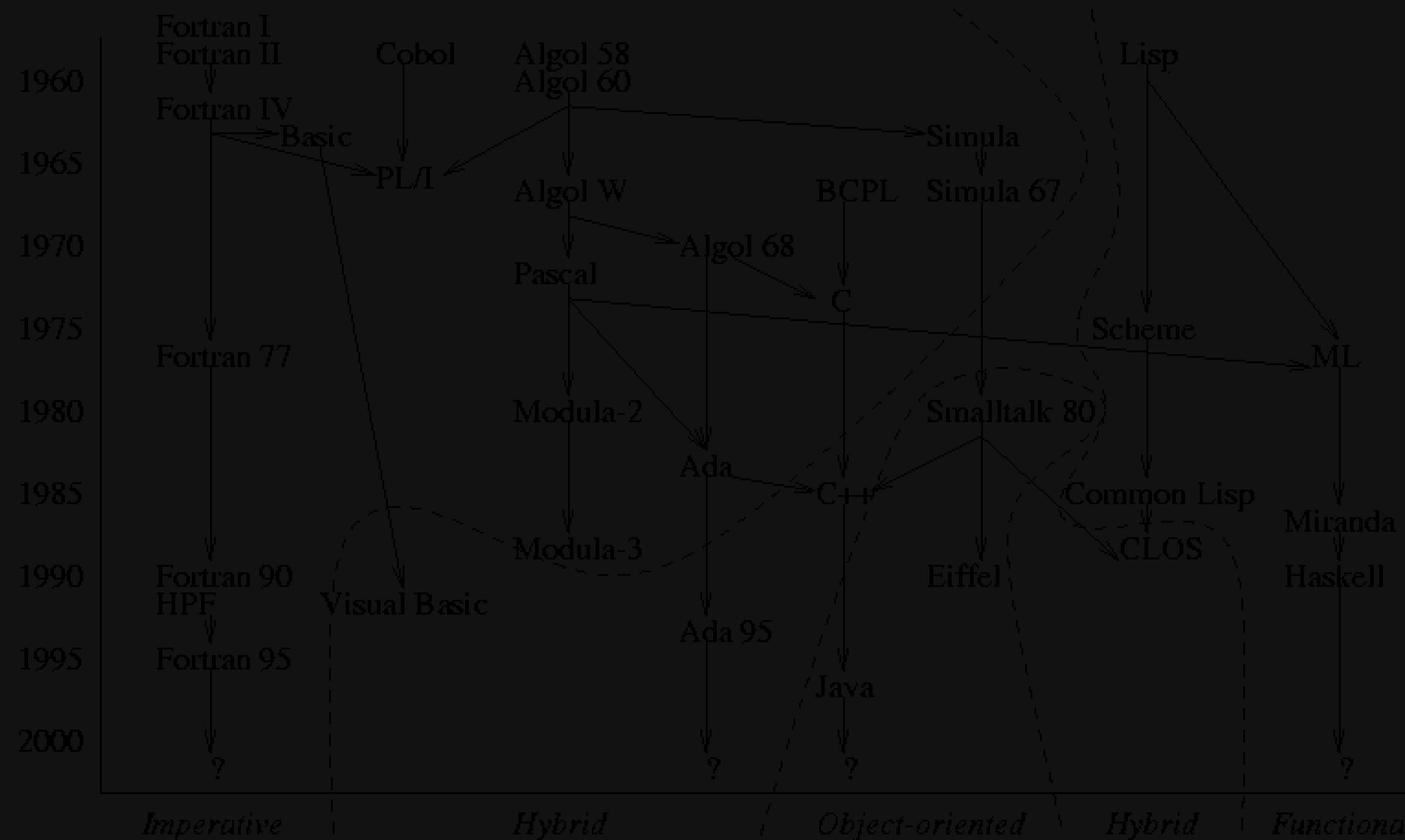
Type System



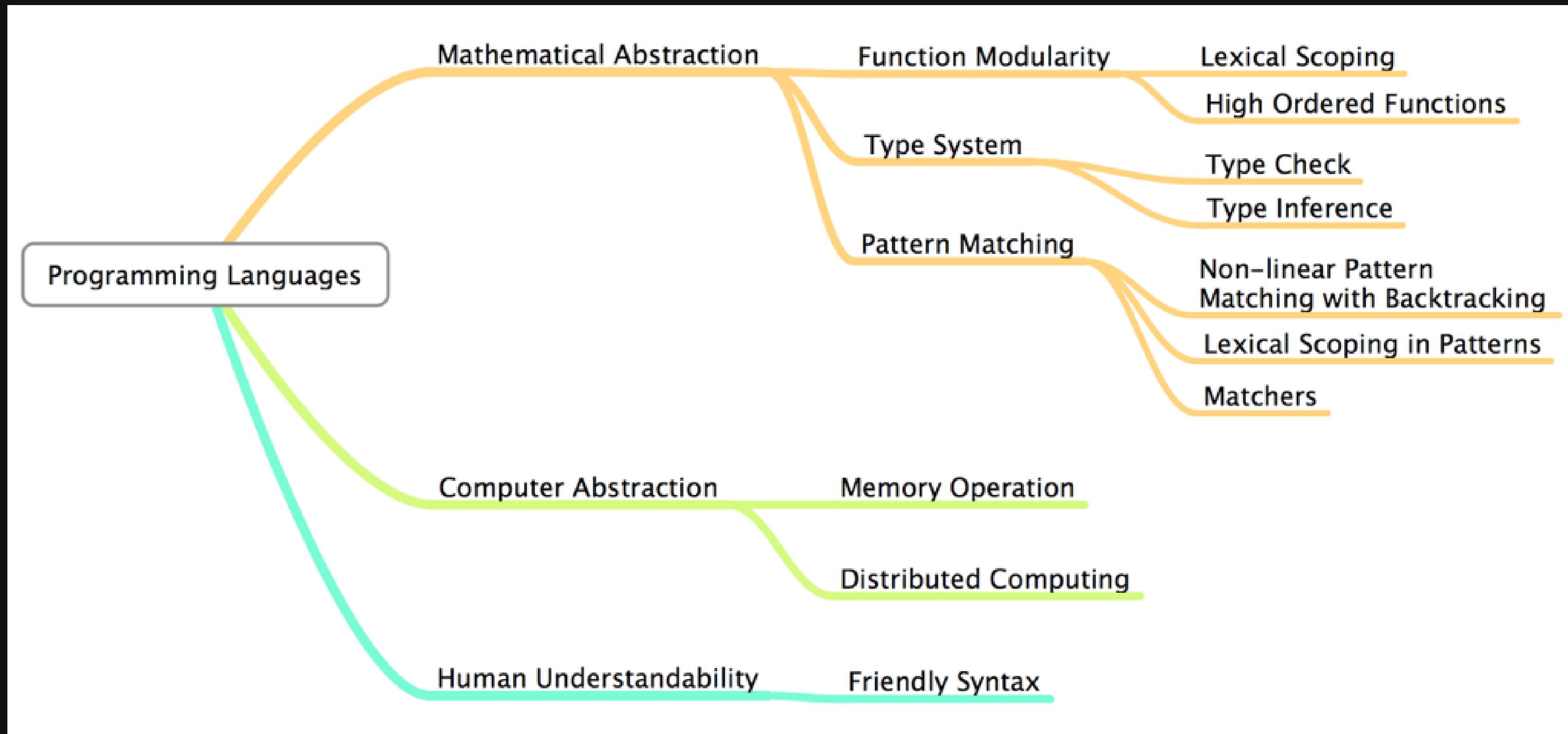
Evolution of Programming Languages

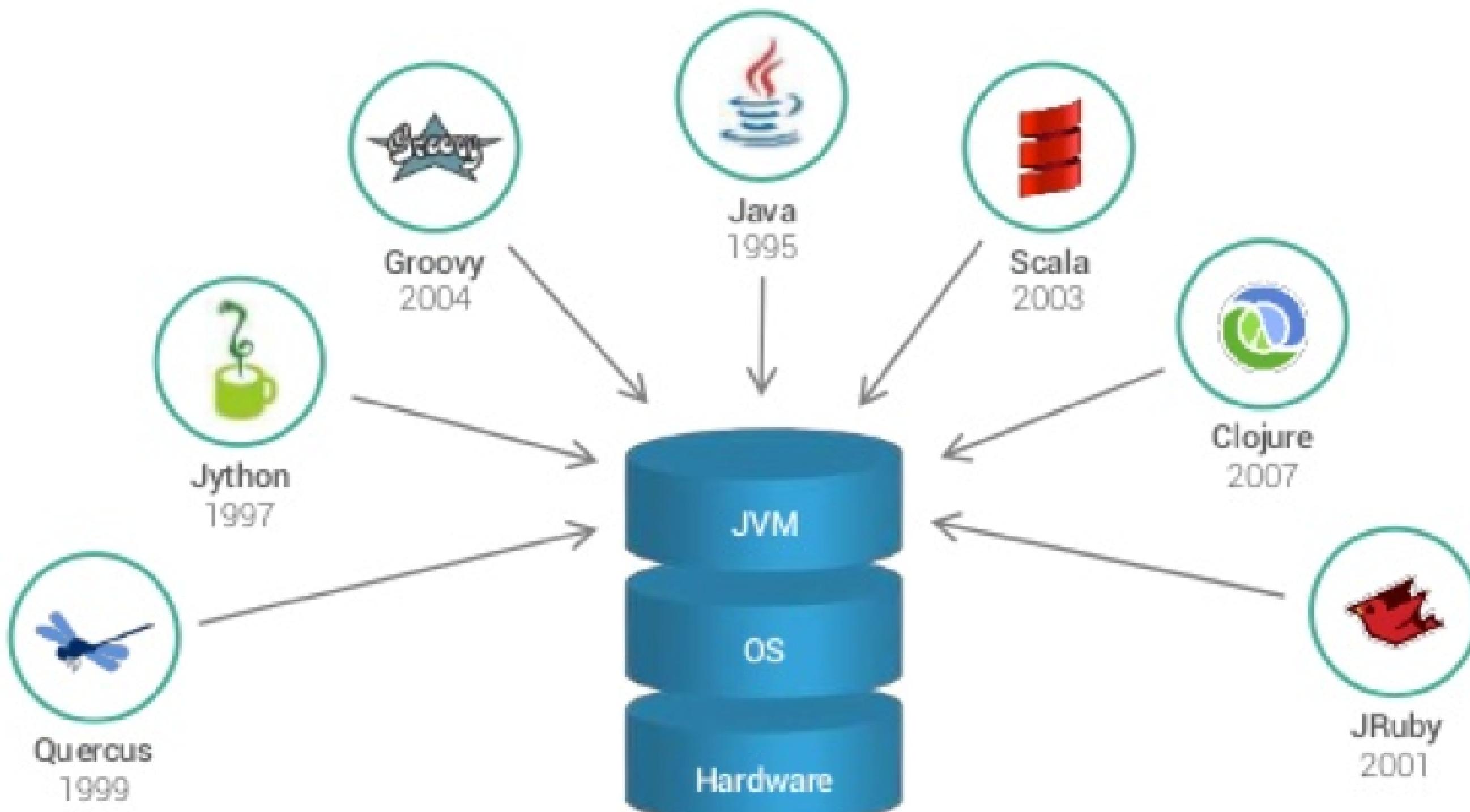


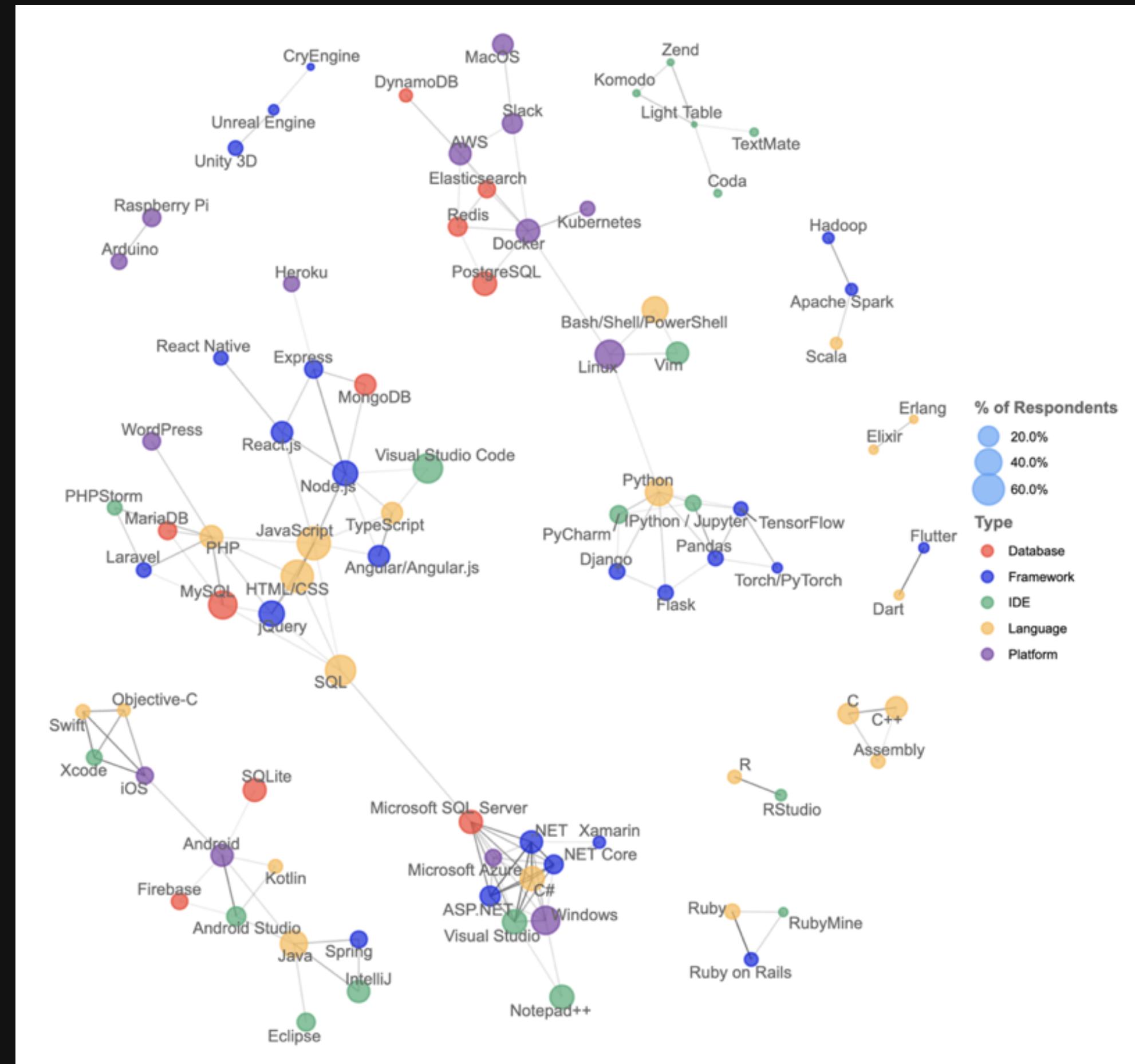
Classify



Evaluate programming languages



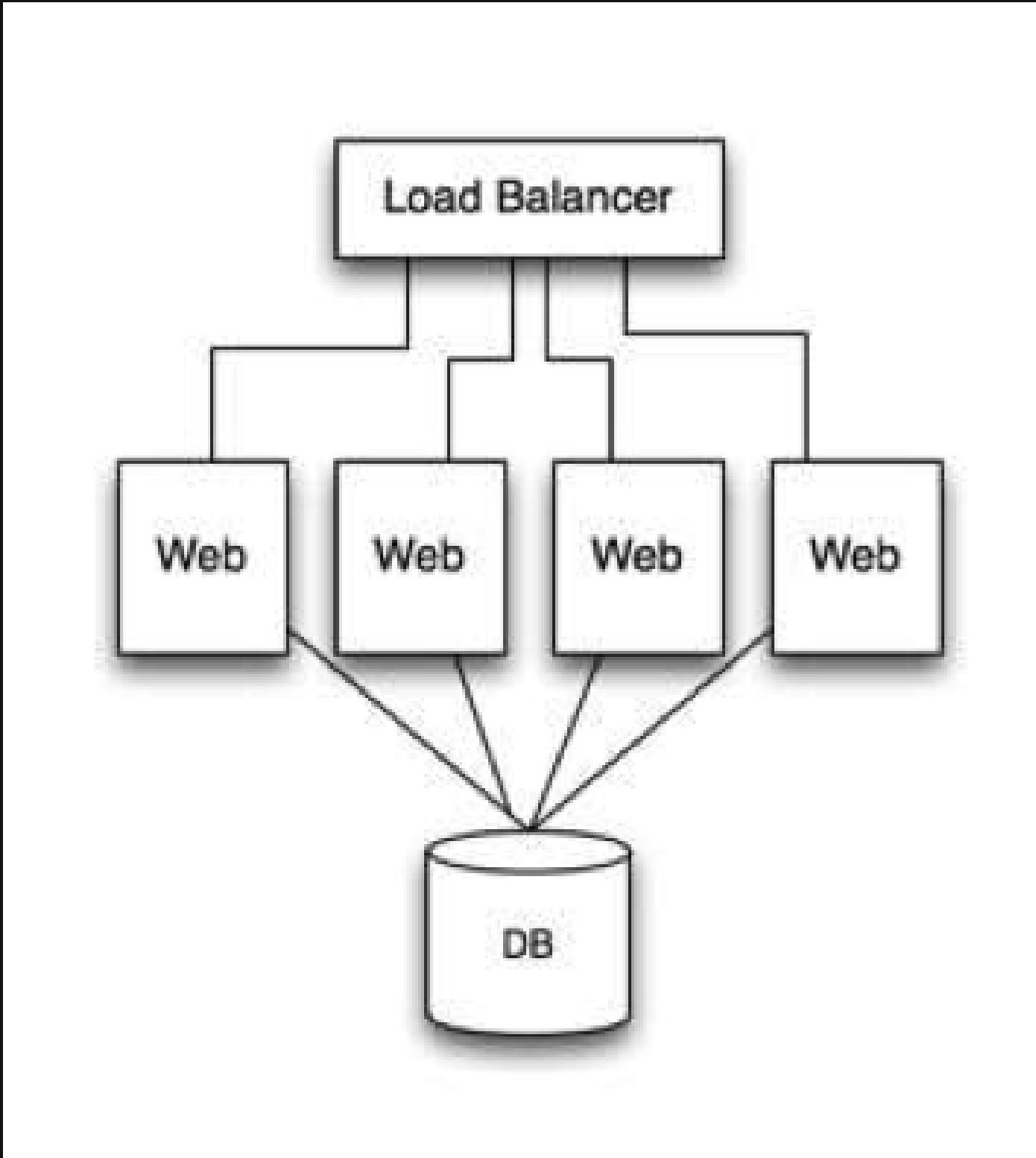




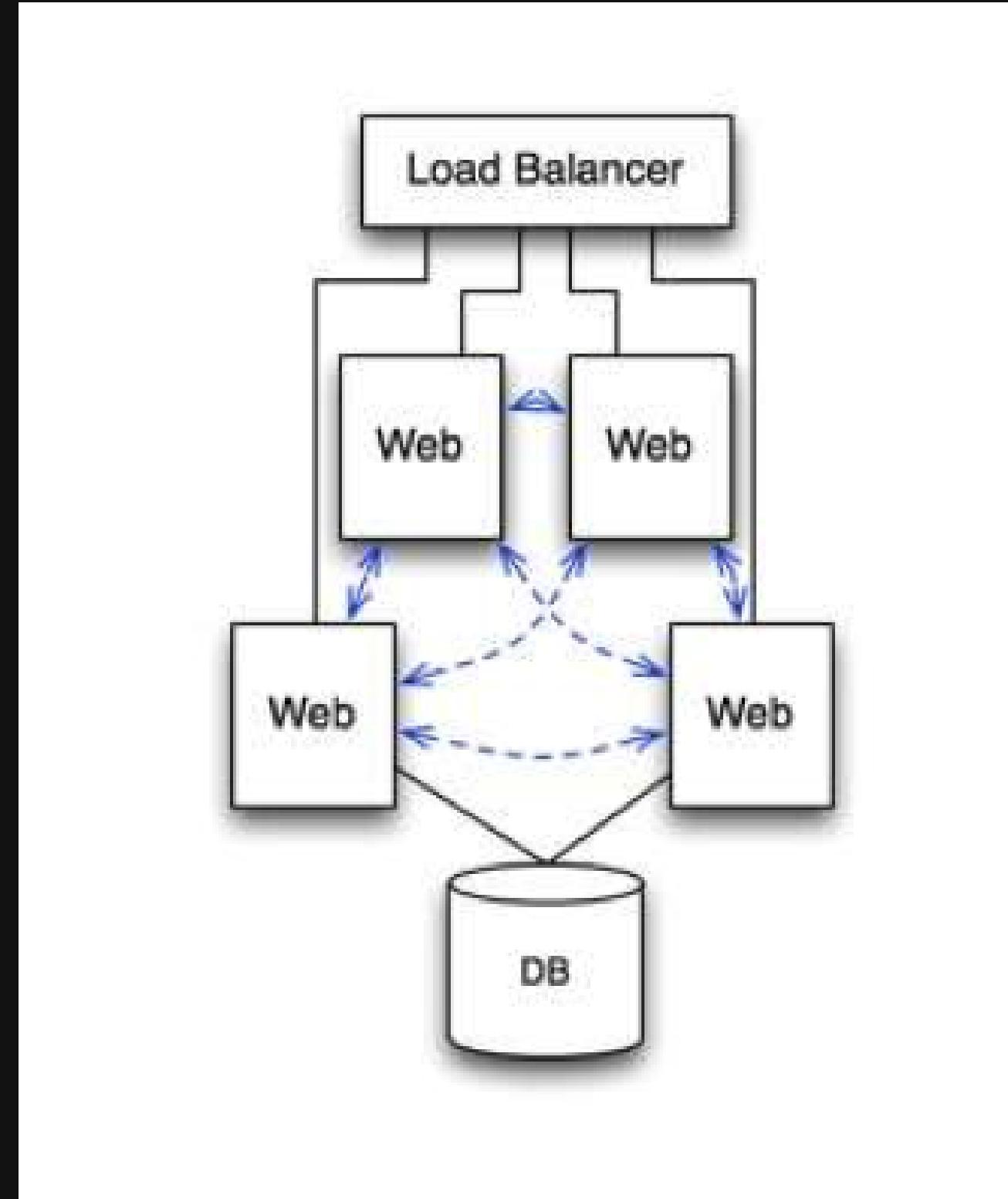
WHAT'S THE BIG DEAL WITH ELIXIR?

- Ruby-like focus on developer productivity
- Embedded database
- Compiles down to code to run on the BEAM Virtual Machine
- BEAM/OTP is what Erlang runs on
- Erlang/BEAM is the best existing language for concurrency, consistency and fault tolerance, hot code swapping
- Erlang does not focus on developer productivity
- Problem in Ruby is concurrency model

Standard Web App



OTP



What's the big deal?

- Facebook paid \$22 billion for WhatsApp
- WhatsApp had \$10 million in revenue
- What was the big deal?
 - Erlang/OTP
 - 2 million users / server
 - No central relay point
 - Scales horizontally
 - Deploys w/o disconnect

Other languages

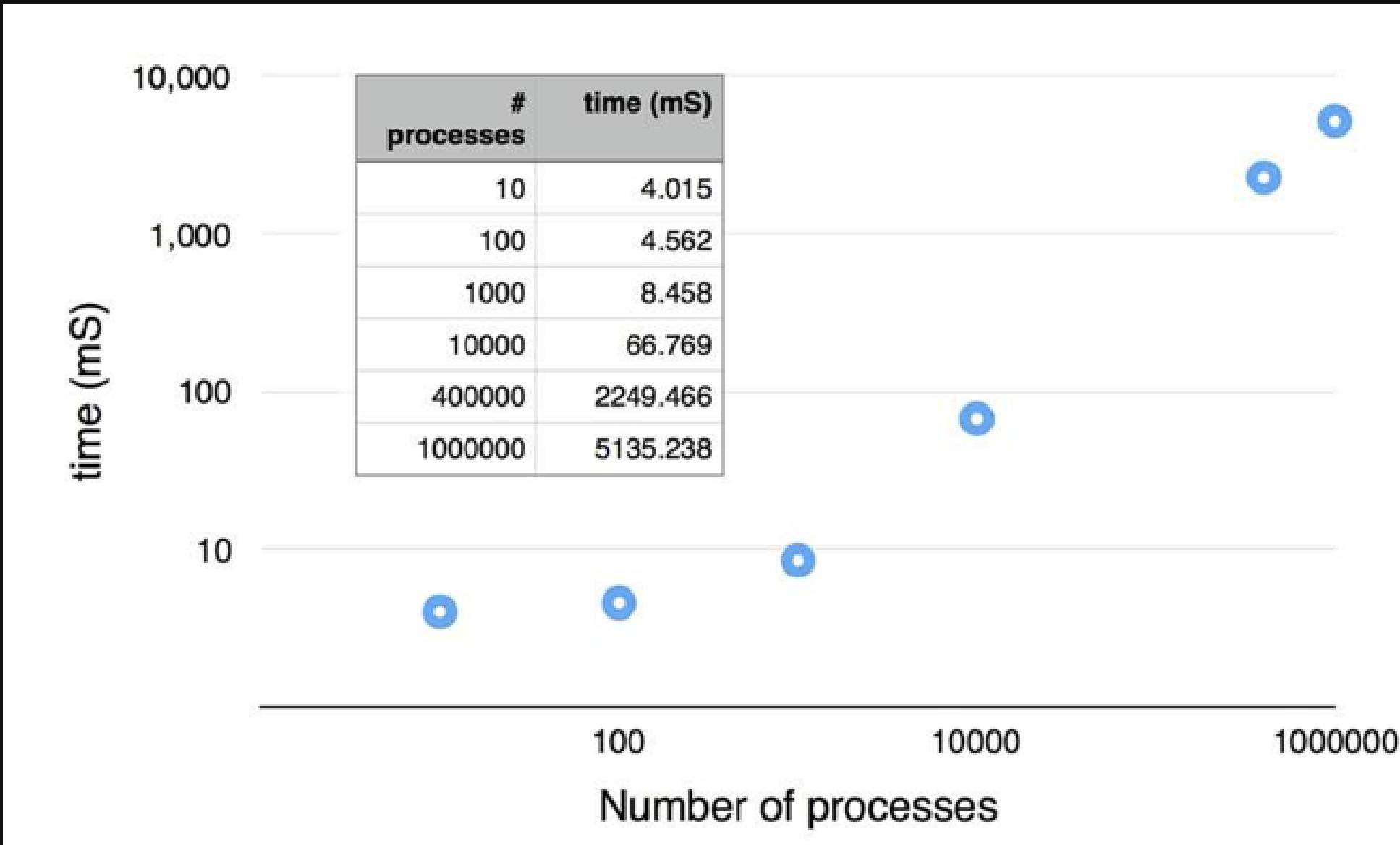
- Boot up
- Memory is shared
 - Where leaks come from
 - Changing shared memory requires a mutex lock
- Garbage collector periodically runs
 - Pause entire stack
- Requests run in threads in the same process
 - Threads are cooperatively scheduled
 - Deployment means shutting down current code, starting new code

Erlang/Elixir/OTP

- No memory is shared
- Data structures are immutable
- Each Erlang process (basically a light thread) has its own HEAP
 - Reclaimed on completion
- Code can be hot deployed
 - New code runs next time it's accessed (existing code keeps running)
- Processes are prescheduled

Sound familiar?

- Difference is size of the allocations
 - An Erlang process is 0.5 kb
 - A Go goroutine is 2 kb (version 1.4)
 - A Java thread is 1024 kb on 64 bit VM
 - PHP request varies by how much is loaded
 - Laravel averages 7-12mb / request



Programming Elixir, Chapter 15 Laptop w/ 4 cores and 4gb of RAM counting concurrently 1,000,000 processes =

- 0.48 gb in Elixir
- 1.91 gb in Golang (go routines)
- 977 gb in Java (threads)