

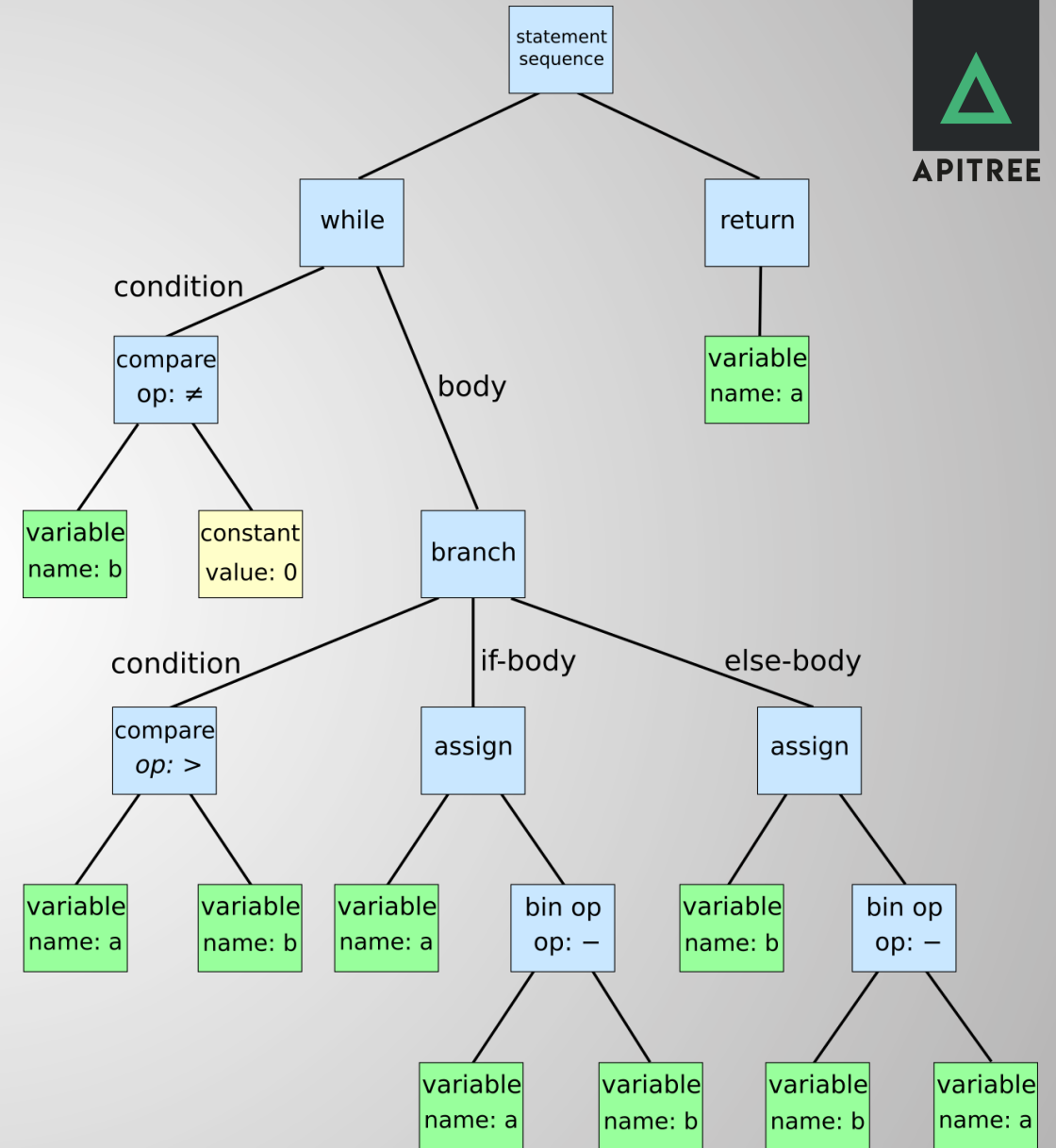


**A P I T R E E**

O AST a Jeho Využití v Projektové Automatizaci

- 🤔 Co jsou ty ASTs
- 💡 Proč by nás to mělo zajímat
- 🤖 Co na to AI

- Abstract Syntax Tree
- Struktura kódu v souvislostech
- Mind map of the code
- Tokeny —> AST objekty
- Žádné kosmetické části, Hard facts



- AST jakožto nedůležitější nástroj
- Příklady využití
  - Kompilátory
    - Nejrychlejší pochopení vypsaneého kódu
  - Statická analýza
    - Lintery (eslint, typescript, ...)
    - Formátování kódu
    - Transpilace (babel)
  - Generace/Transformace kódu (= Codegens/Codemods)

- Codemods & Codegens 💪
- Automatizovaná migrace z jednoho zápisu na druhý
  - Nová breaking verze, jiná library nebo úplně jiný jazyk
- Automatizované udržování codebase podle daných šablon
  - Tvorba nových projektů už jen drobností
  - Přechod na nové verze bez námah
  - Změna build systému jenom otázkou jednoho updatu
  - Snížení complexity udržování několika projektů najednou
  - Jasnější definice projektů / Unifikace

- Plno nástrojů na statickou analýzu nebo využití statické analýzy
  - >1 standardů
  - >1 parserů
- Node.js bez AST API (= C#, Java)
- PHP má svoje AST API
- Typescript podobně jako u PHP

## -Estree

- Přímý ECMA Script standard
- Jeden z prvních
- Převzato z API SpiderMonkey

## -Shift

- Podobně jako Estree
- Není oficiálním standardem
- Spíše orientován k transformaci

## -Babel

- Není oficiálním standardem
- Jako první obsahuje proposals



- Typescript
- Esprima
  - Jeden z prvních, jeden z nejrozšířenějších
- Babel
  - @babel/parser na statickou analýzu
- Espree
  - ESLint





# AST Explorer

- JSCodeShift
  - Migrační nástroj od Facebooku
  - Jasná struktura definice
  - Širší spektrum dostupných parserů zastřešené jedním API
  - Široce využívaný (MUI)
- NX
  - projektová automatizace na dlani
  - Orchestrace monorep
    - Script cache, distributed script execution, project map, ...
    - Nx plugins
  - Zralá komunita, obrovské množství integrací (skrz nx plugins)
  - Není limitováno pouze JS monorep (c#, rust, go)
- tsuru
  - Jednoduchá a jasná definice CLI aplikací (příkazy, šablony, ...etc)
  - ESM + CJS podpora
  - Typesafe šablony
  - Možnost vlastního pojmenování



JSCodeShift

- JSCodeshift
  - Migrační nástroj od Facebooku
  - Jasná struktura definice
  - Širší spektrum dostupných parserů zastřešené jedním API
  - Široce využívaný (MUI)
- NX
  - projektová automatizace na dlani
  - Orchestrace monorep
    - Script cache, distributed script execution, project map, ...
    - Nx plugins
  - Zralá komunita, obrovské množství integrací (skrz nx plugins)
  - Není limitováno pouze JS monorep (c#, rust, go)
- tsuru
  - Jednoduchá a jasná definice CLI aplikací (příkazy, šablony, ...etc)
  - ESM + CJS podpora
  - Typesafe šablony
  - Možnost vlastního pojmenování



NX

- JSCodeshift
  - Migrační nástroj od Facebooku
  - Jasná struktura definice
  - Širší spektrum dostupných parserů zastřešené jedním API
  - Široce využívaný (MUI)
- NX
  - projektová automatizace na dlani
  - Orchestrace monorep
    - Script cache, distributed script execution, project map, ...
    - Nx plugins
  - Zralá komunita, obrovské množství integrací (skrz nx plugins)
  - Není limitováno pouze JS monorep (c#, rust, go)
- Tsuru
  - Jednoduchá a jasná definice CLI aplikací (příkazy, šablony, ...etc)
  - ESM + CJS podpora
  - Typesafe šablony
  - Možnost vlastního pojmenování

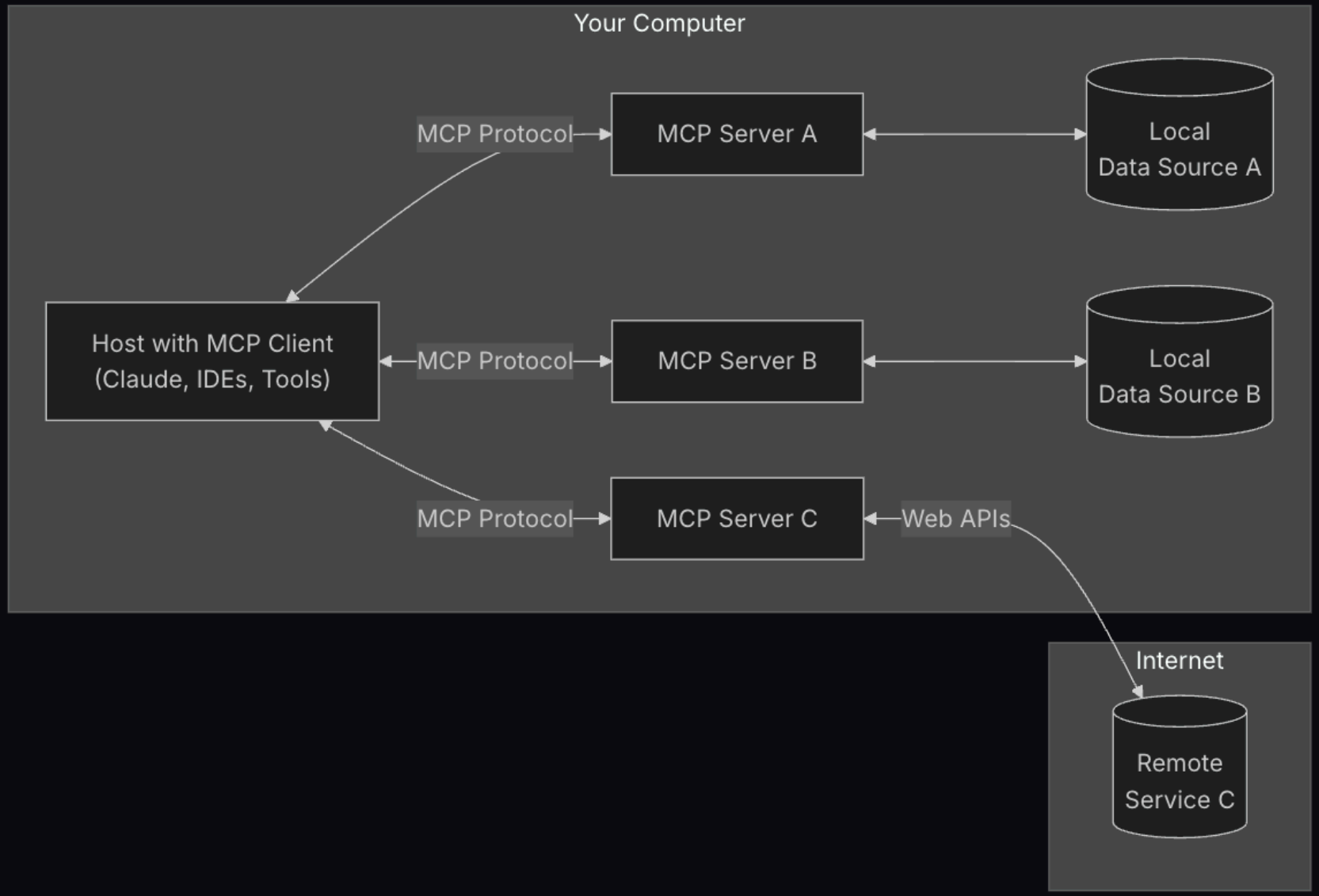


Tsuru

- Horší prosazení ohledně kontribucí
- Při integraci je potřebné využívat automatické dependency updates (renovate bot)
  - Autocomplete když projde validace
- Verze závislostí řešit centrálně, případně povolit range
- Testy jsou přímo nutností
- Windows vs Unix (cesty, fs, ...)
- Při jasné definici: zefektivnění údržby a tvorby nových projektů +40%
- Jednodušší šablony na začátku -> upravovat podle potřeb
  - Ejs nebo string literal jako jednoduchou statickou šablonu
  - AST posléze když je potřeba dynamických šablon
    - Typescript nebo shift parser
    - Babel má moc metadat, ale jednodušší API
- Využití komunitních šablon způsobuje neřízený tlak
- Je dobré si vytvořit šablony na často vznikající části projektu
  - Automatické změny v dockerfile, pipelines, propojení s dalšími částmi projektové definice (terraform)



- Existující projekty na generaci projektů
- Částečná možnost projektové správy
- Neuspokojivé/Nepřesné výsledky
- Master Proompting required
- Anthropic Model Context Protocol (MCP)
  - Jednodušší user interface
  - Přesnější výsledky



**Query:** "If I change the public API of feat-product-detail, which other projects might be affected by that change?"

Nx Graph — aishop

Chat

Based on the information I've gathered, I can now determine which projects might be affected by changes to the public API of feat-product-detail.

From the Nx workspace information, I can see that:

1. The `@aishop/feat-product-detail` project has dependencies on:
  - `@aishop/data-access-products`
  - `@aishop/ui-product-detail`
2. The projects that depend on `@aishop/feat-product-detail` are:
  - `@aishop/shop` (the main application)

If you change the public API of `@aishop/feat-product-detail`, the following projects would be directly affected:

1. **@aishop/shop** - This is the main application that imports and uses the `feat-product-detail` library. Any changes to the public API would require corresponding updates in how the shop app consumes this feature.

It's worth noting that:

1. The `@aishop/feat-product-detail` project is tagged as `npm:public`, which means it's intended for public consumption.

Generating. Stop



MCP + Local LLM

# THE RT OF TECHNOLOGY\_

APITREE.CZ

@ info@apitree.cz

+420 607 665 681

Francouzská 75/4, 120 00 Praha 2