

# **Forthic**

## **Forth Day 2024**

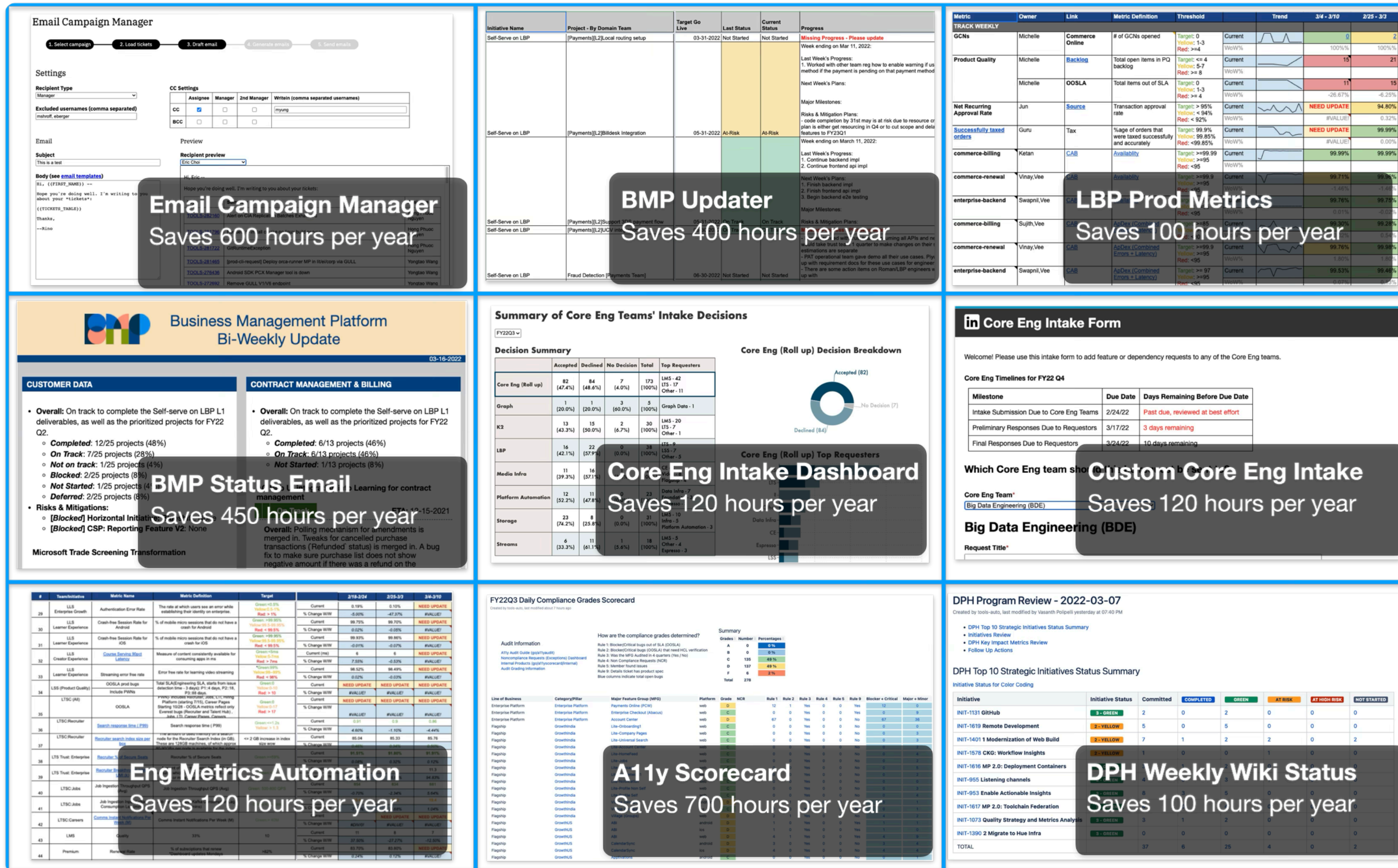
**Rino Jose**  
**Nov 16, 2024**

# 1 - Motivation for Forthic

- Needed a way to build lots of custom web-based project tools at LinkedIn
- Wanted to use Forth, but struggled with strings and JSON data
- Decided to build something Forth-like, but focused on the top of the app



<https://github.com/linkedin/forthic>



These could be built quickly: 100 per year. Over 5 years, there were over 800 of these Forthic apps and automations

# 2 - Incomplete by design

Requires a host language to run on top of (any will do)

folder forthic-py

folder forthic-react/v1

main

forthic / experimental /

rjose V4 (#14)

| Name          |
|---------------|
| ..            |
| forthic-cpp   |
| forthic-hs    |
| forthic-jl    |
| forthic-nvcc  |
| forthic-rs    |
| forthic-swift |
| forthic-zig   |

main

forthic / experimental / pre-forthic /

rjose V4 (#14)

| Name       |
|------------|
| ..         |
| forrth-asm |
| forrth-cs  |
| forrth-erl |
| forrth-f90 |

## 2 - Incomplete by design

### No "if" statements or loops

Instead, it uses arrays and records and high level words like MAP and GROUP-BY to manipulate data

#### DEMO: MAP over array and record

1. Clone Forthic Repo
2. Run make
3. Go to <http://localhost:8000/talks/map/>



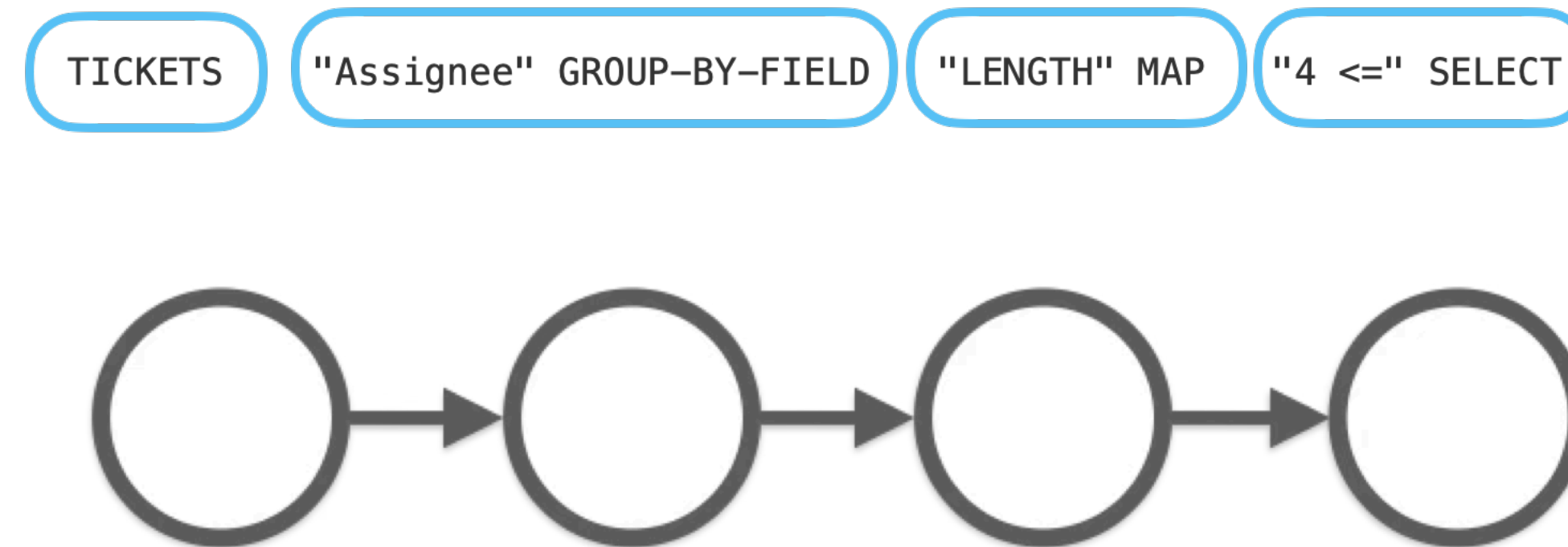
## 2 - Incomplete by design

Implication: Encourages straight-line, one-line coding

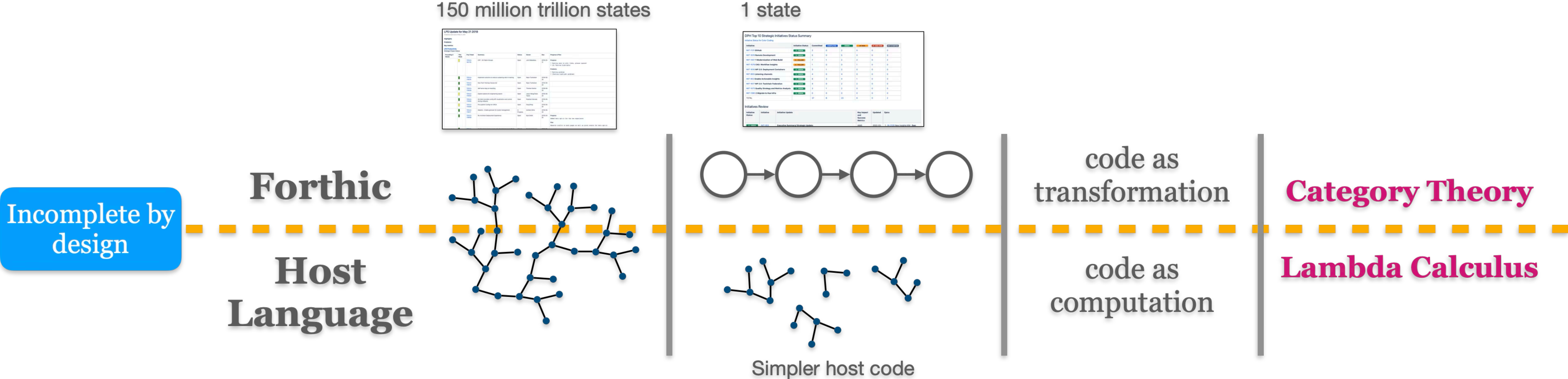
```
: JQL      "resolution = null and updated > -30d";  
: FIELDS   ['Summary' 'Assignee' 'Status'];  
: TICKETS  JQL FIELDS jira.SEARCH;  
  
TICKETS "Assignee" GROUP-BY-FIELD "LENGTH" MAP "4 <=" SELECT
```

## 2 - Incomplete by design

Implication: Code looks like composition of transformations



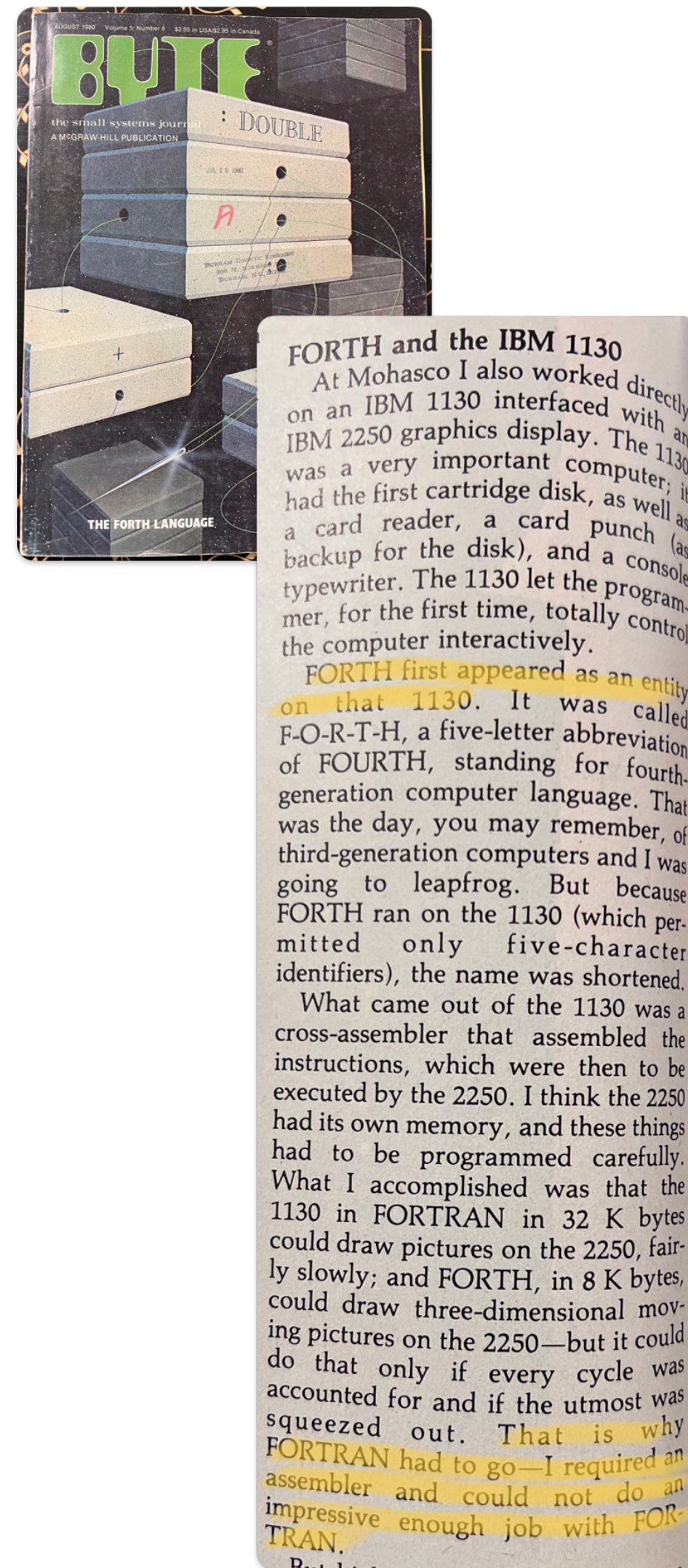
# 2 - Incomplete by design





# 3 - Host languages open up everything

First FORTH was built in FORTRAN



Host languages have more capabilities now

**DEMO: Pull Jira issues and manipulate JSON**

1. Clone Forthic Repo

2. Run make

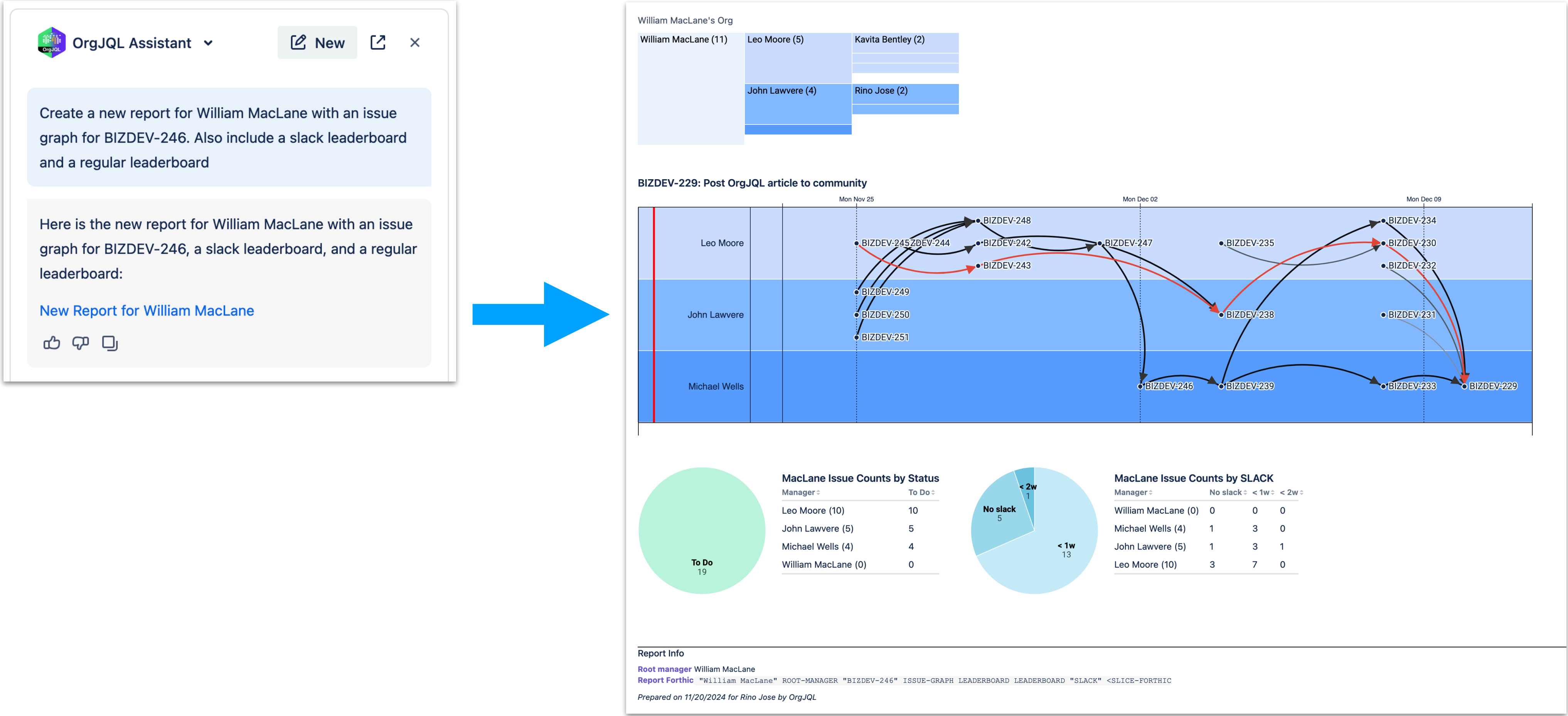
3. Go to <http://localhost:8000/talks/jira/>

*NOTE: You'll need a Jira account and some data :-)*



# 4 - Forthic can extend apps after deployment

Example: AI can extend Forthic apps



One line Forthic App: "William MacLane" ROOT-MANAGER "BIZDEV-246" ISSUE-GRAPH LEADERBOARD LEADERBOARD "SLACK" <SLICE-FORTHIC

## 4 - Forthic can extend apps after deployment

Example: AI understands Forthic apps (mostly)



Run Forthic [1 2 3] "3 \*" MAP ["a" "b" "c"] SWAP ZIP  
REC

The result of the operation is:

```
1 [
2   {"a": 3},
3   {"b": 6},
4   {"c": 9}
5 ]
```



Explain this

Certainly! Let's break down the Forthic code step by step:

1. [1 2 3] "3 \*" MAP :
  - This takes the list [1, 2, 3] and applies the operation 3 \* to each element.
  - The result is [3, 6, 9] .
2. ["a" "b" "c"] :
  - This pushes the list ["a", "b", "c"] onto the stack.
  - The stack now looks like this: [3, 6, 9] ["a", "b", "c"] .
3. SWAP :
  - This swaps the top two elements on the stack.
  - After the swap, the stack looks like this: ["a", "b", "c"] [3, 6, 9] .
4. ZIP :
  - This combines the two lists into pairs: [{"a", 3}, {"b", 6}, {"c", 9}] .
5. REC :
  - This converts each pair into a dictionary where the first element of the pair is the key and the second element is the value.
  - The result is [{"a": 3}, {"b": 6}, {"c": 9}] .

So, the final result is a list of dictionaries where each key from the list ["a", "b", "c"] is paired with the corresponding value from the list [3, 6, 9] .



Not quite. The final result is just a dictionary (record)