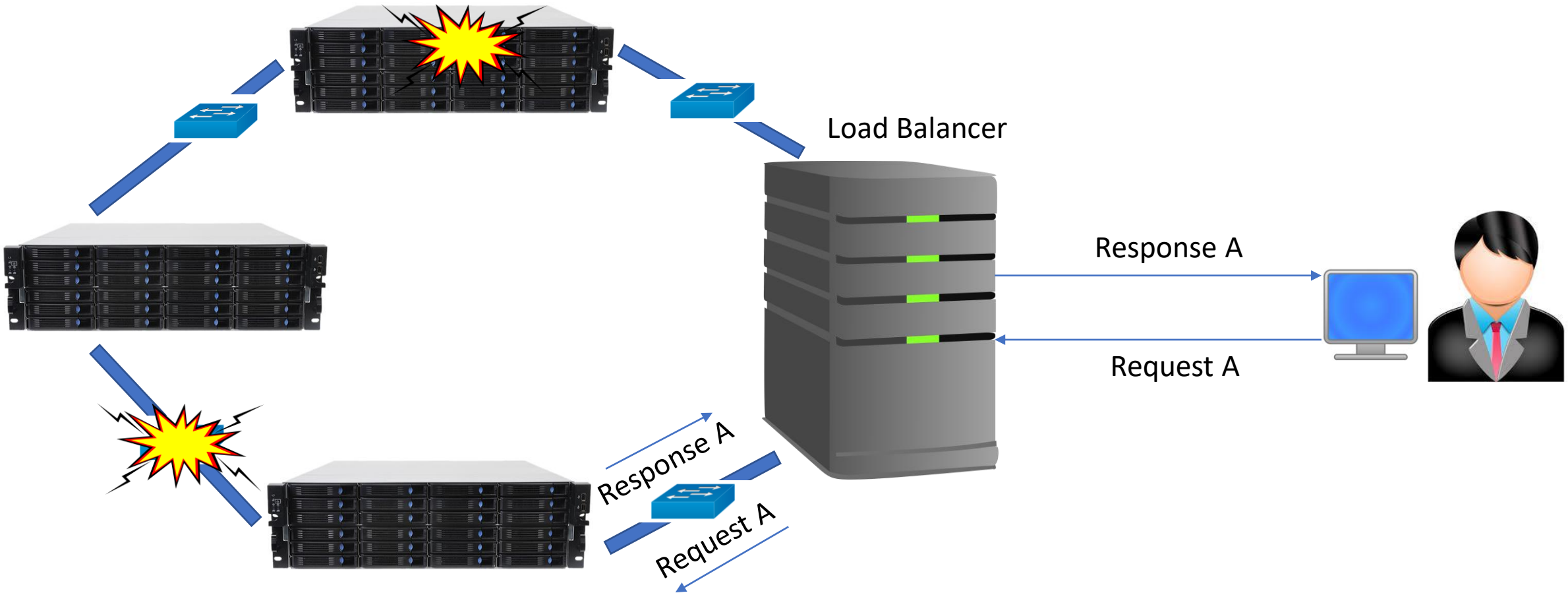
A System-Wide Debugging Assistant Powered by Natural Language Processing

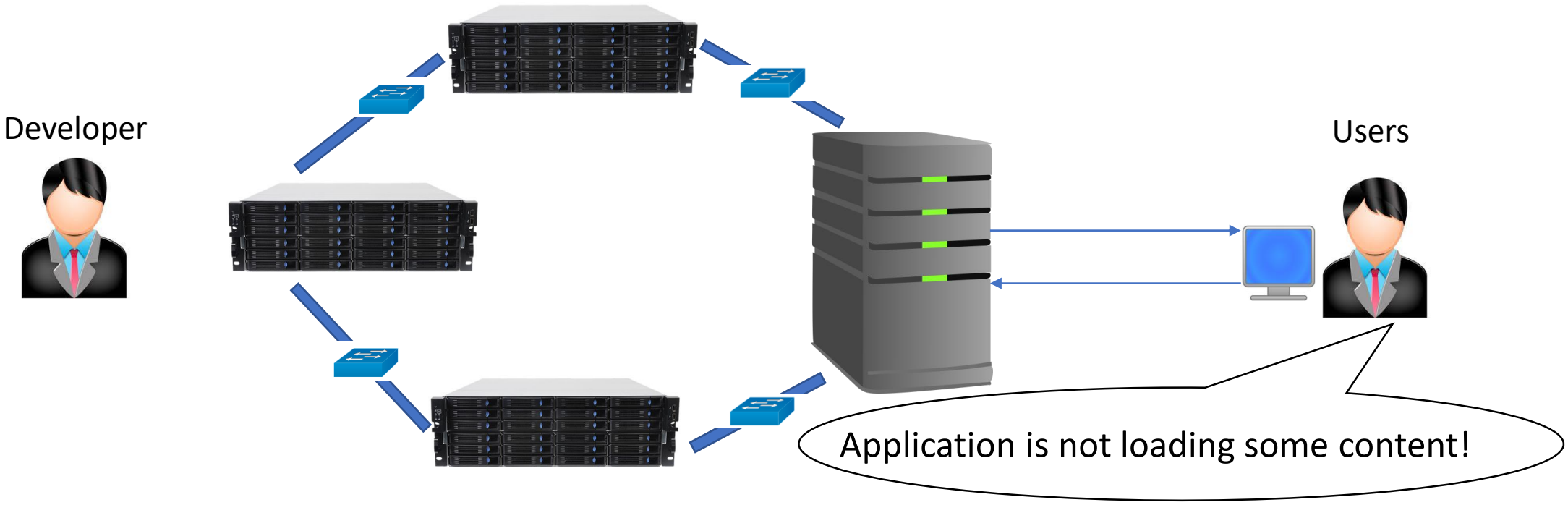
Pradeep Dogga* Karthik Narasimhan† Anirudh Sivaraman‡ Ravi Netravali*

UCLA * PRINCETON UNIVERSITY † NYU ‡

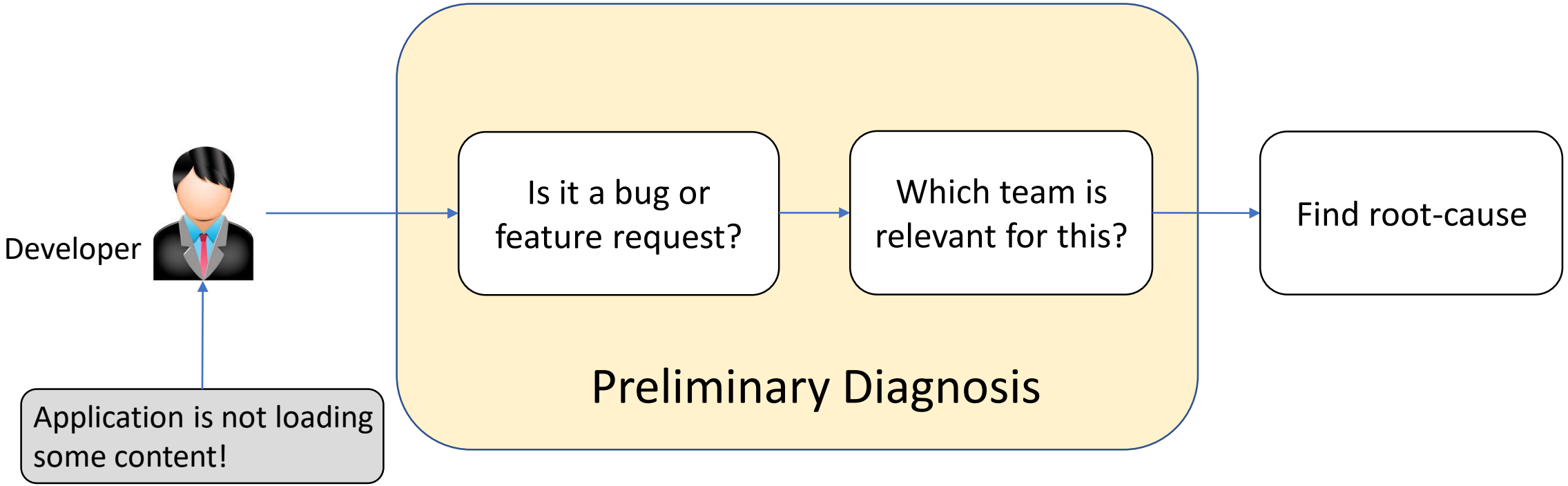
Distributed Systems are complex



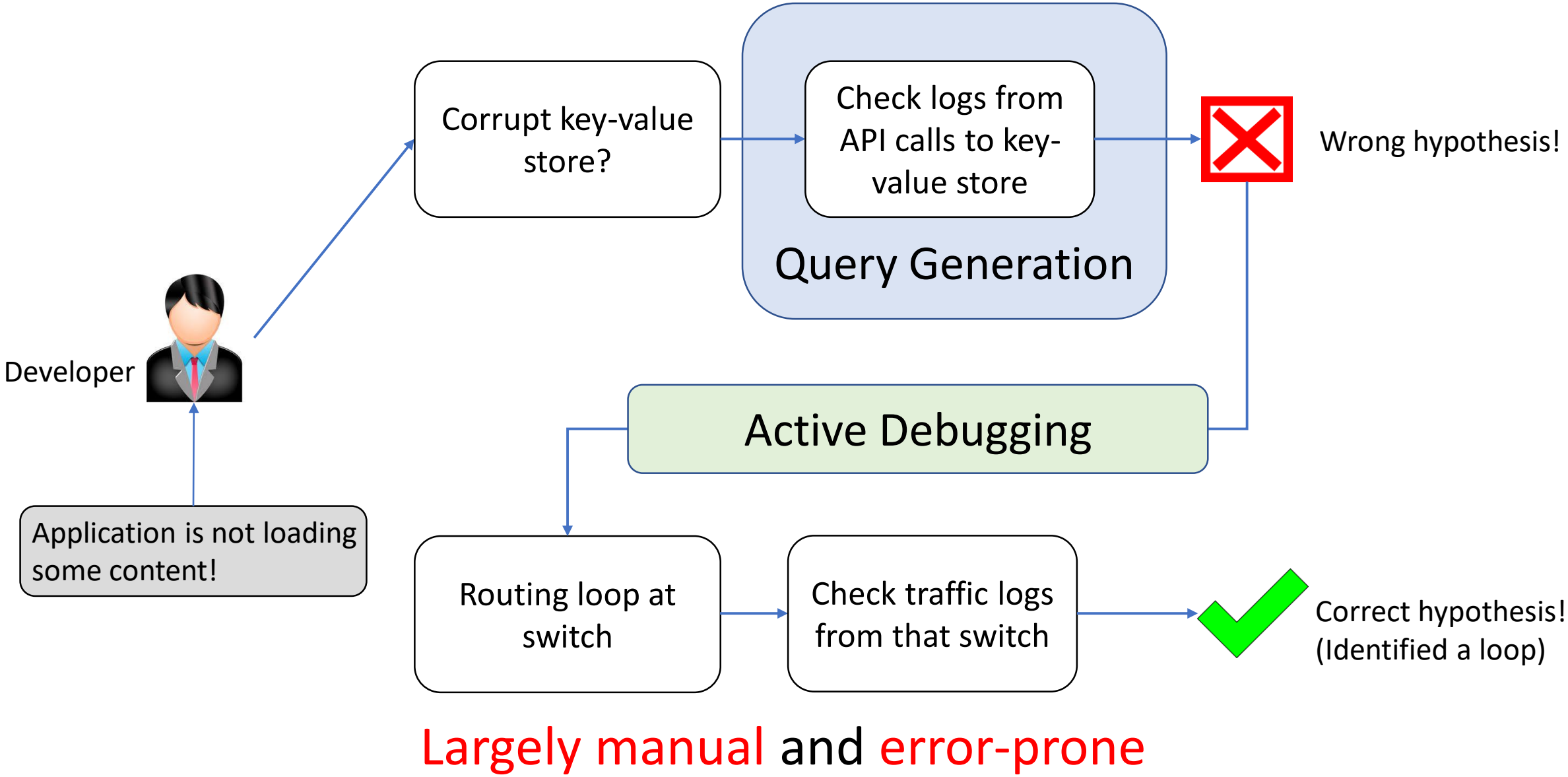
Debugging is hard - abstraction gap



Painful debugging process

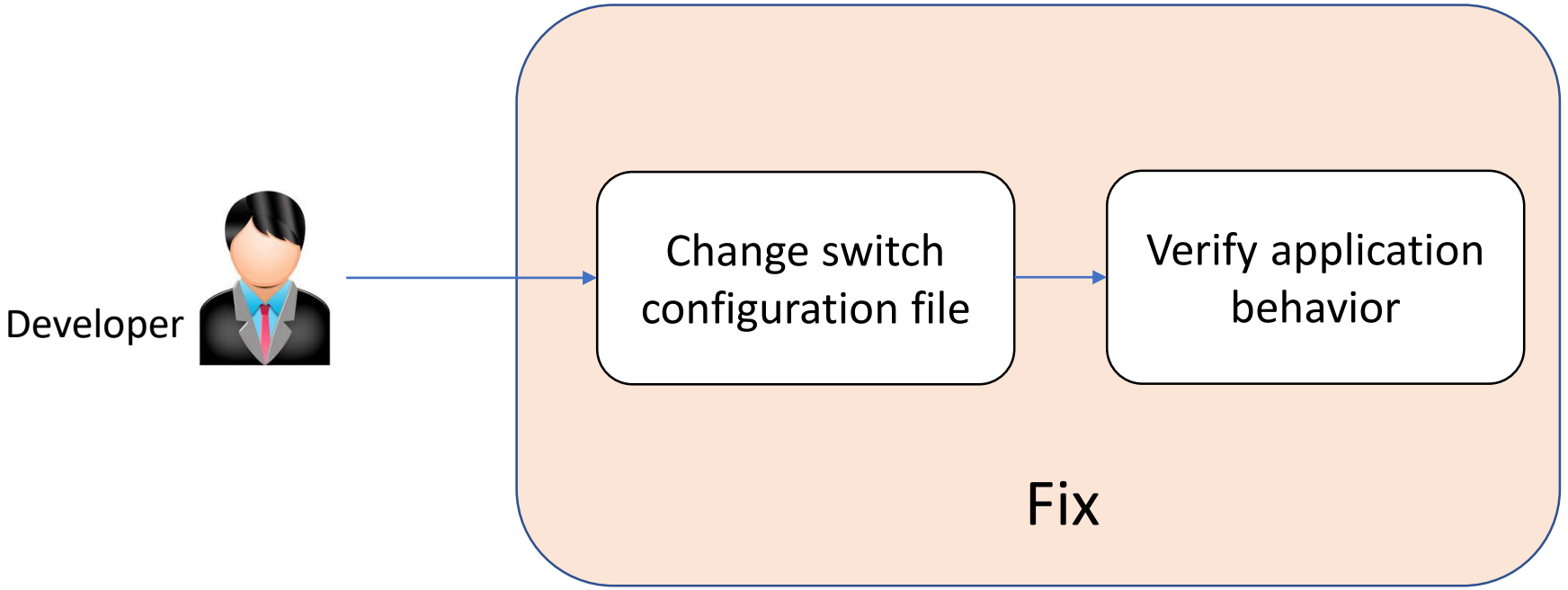


Painful debugging process – Finding root cause

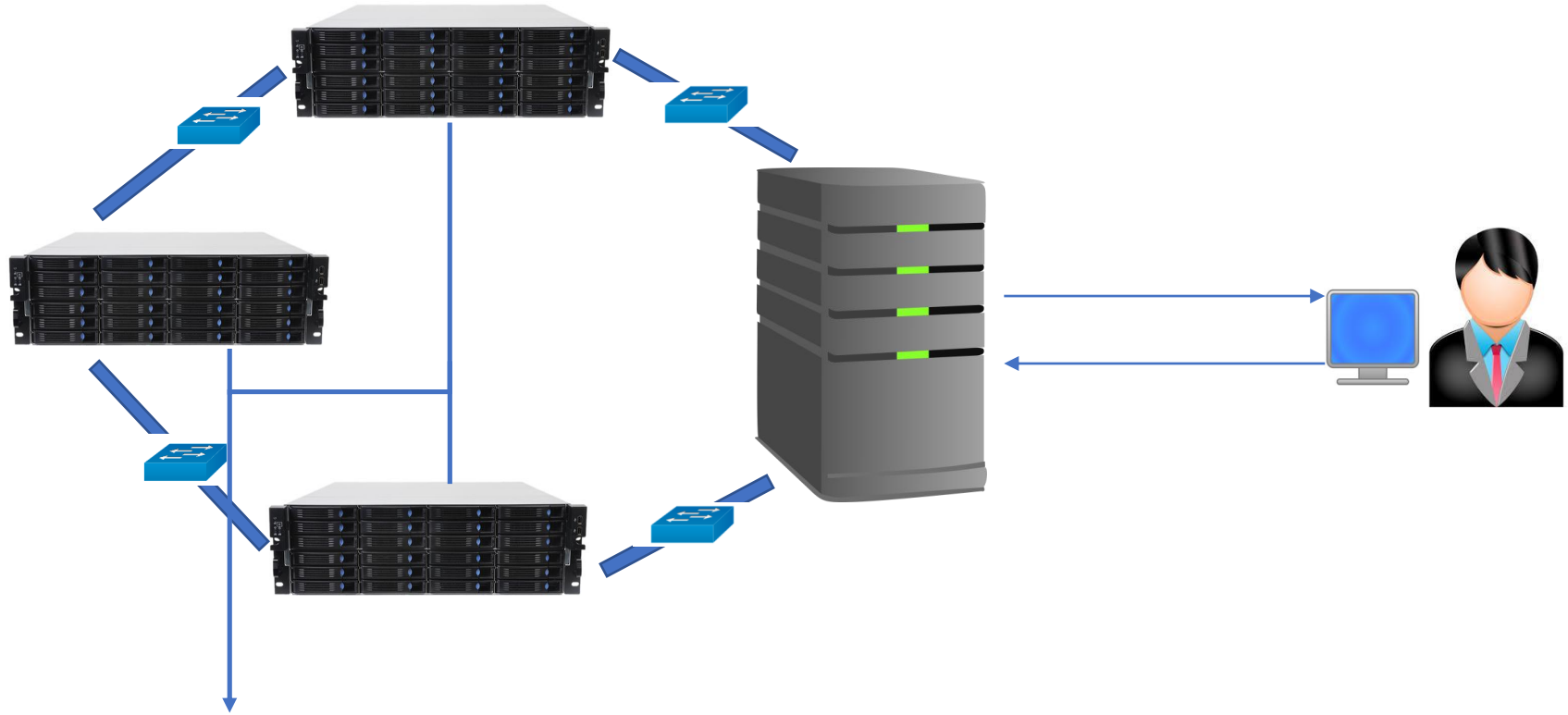


Largely manual and error-prone

Painful debugging process – Generate Fix



Systems debugging tools

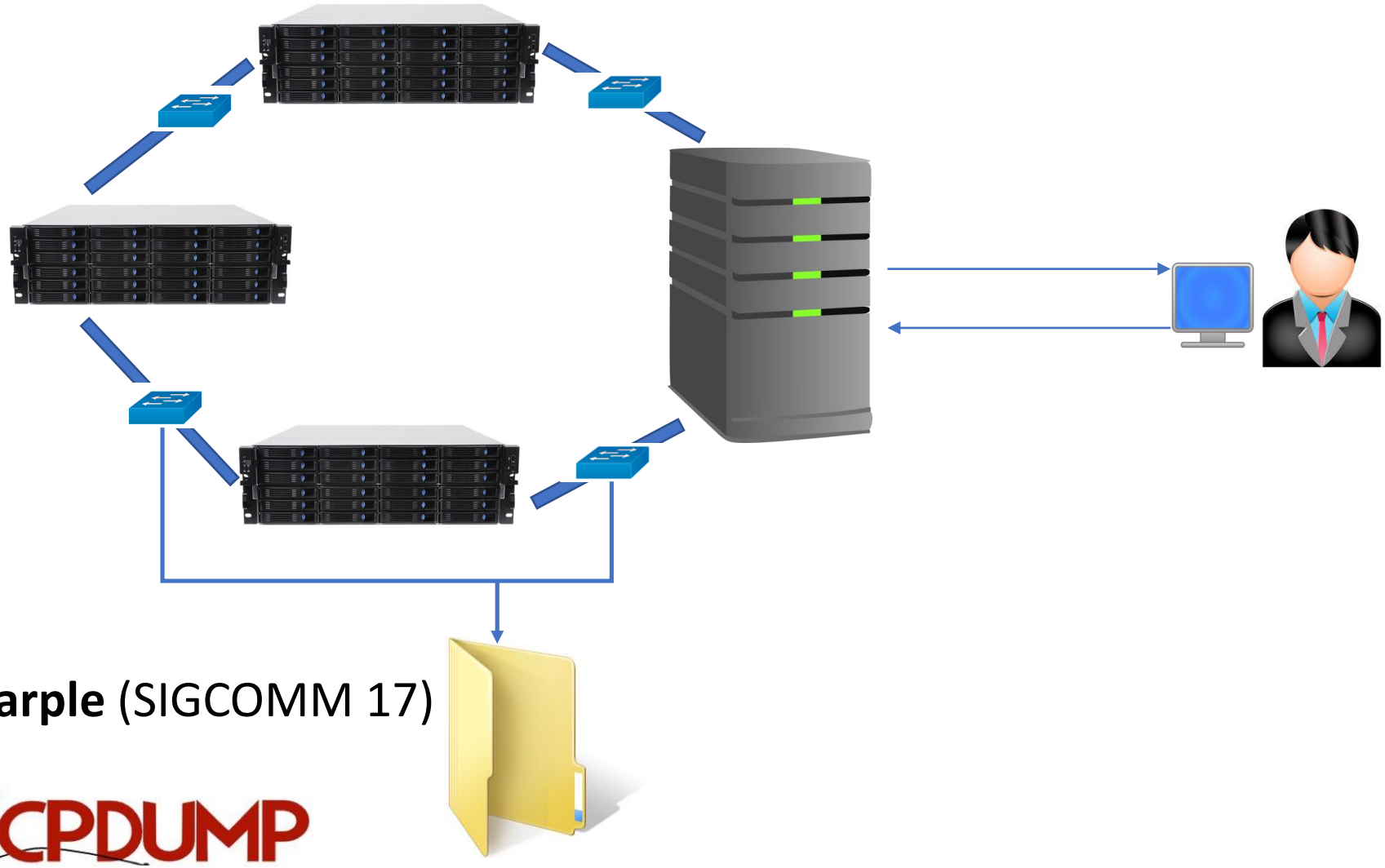


OPENTRACING

JAEGER

Application Logs

Systems debugging tools

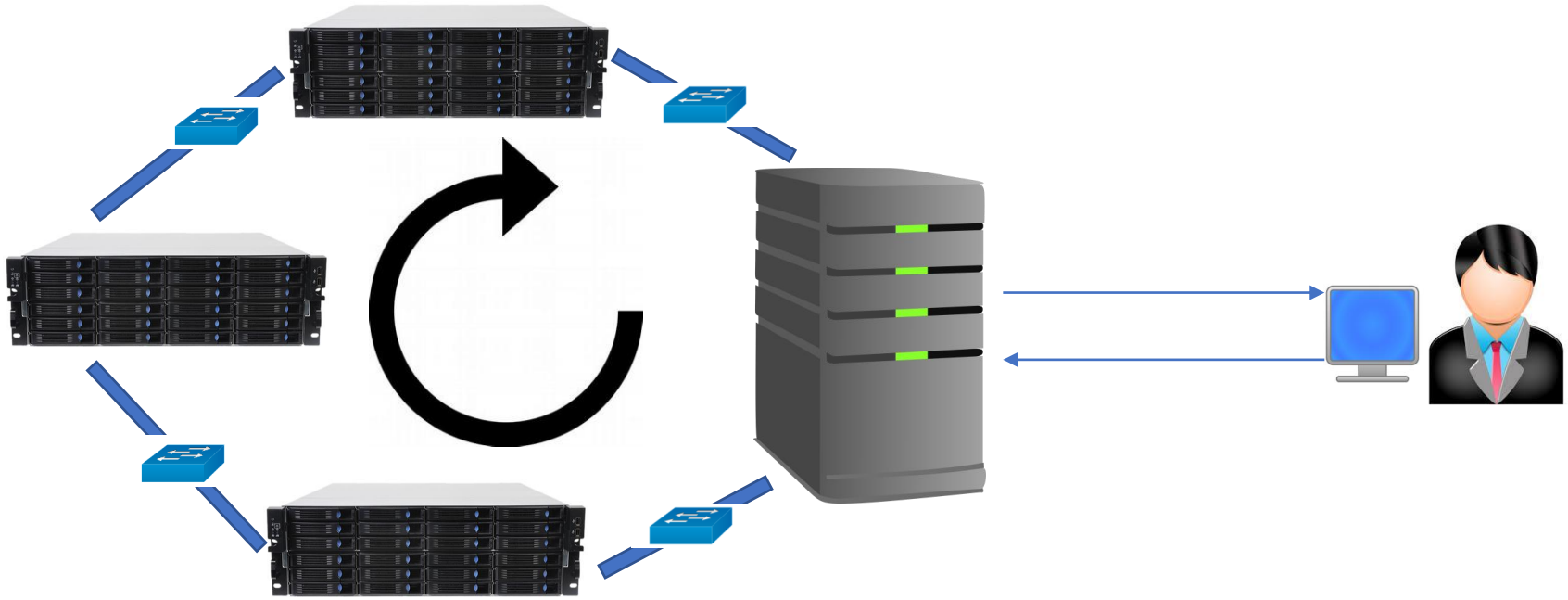


Marple (SIGCOMM 17)

TCPDUMP

Network Metrics

Systems debugging tools



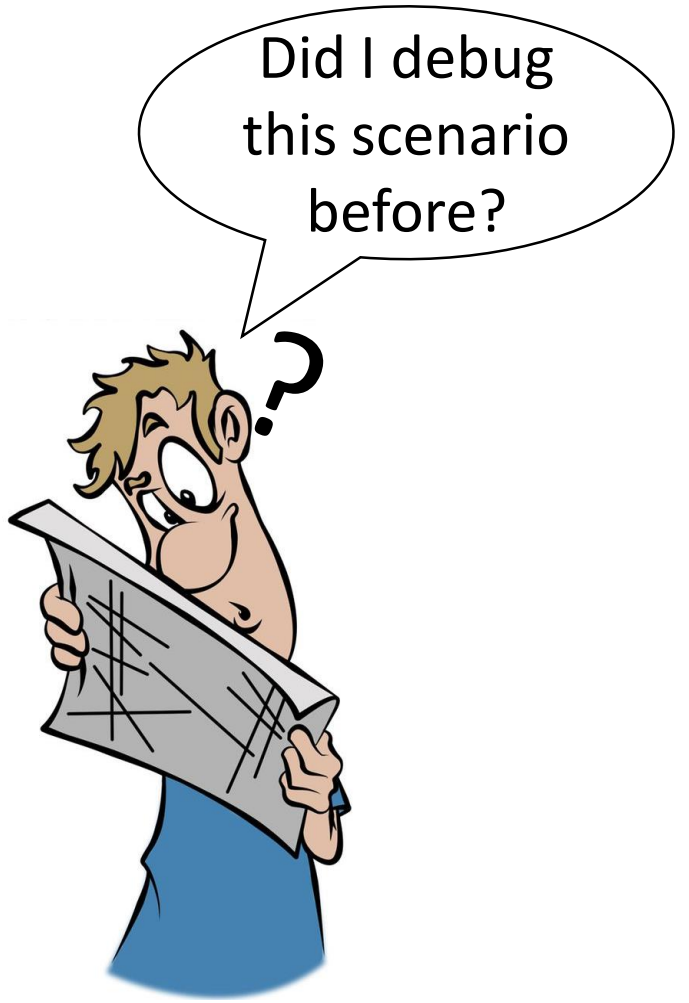
Canopy (SOSP 17)
Pivot Tracing (SOSP 15)

Distributed systems tracing

Debugging remains difficult

- Still manual and error-prone:
  - Which tool?
  - When?
  - How?

- Debugging intuitions are hard-won!



Can we use a data-driven approach to automate steps in end-to-end debugging?

Large amounts of debugging data

Distributed tracing at Pinterest with new open source tools

The New Stack
@thenewstack

To monitor its thousands of services, Facebook captures about billion traces a day (about ~100TB collected), a dynamic sampling of the total number of interactions per day — @Facebook's Haozhe Gao and Joe O'Neill #QConNYC

The New Stack
thenewstack

u find a new type of performance issue, the temptation
a new set of metrics to a dashboard. Most of the time
t a good idea. Overly busy dashboards can quickly lead
ive overload — Google's @lizthegrey on #microservices

Ticketmaster Traces 100 Million Transactions per Day with Jaeger

Orate Follow
May 6 · 6 min read

Two big classes of data:

Quantitative/Structured	**Unstructured/Natural Language**
Logs from tools	User Issues
Performance metrics	Documentation and comments
Source code	Past bug reports

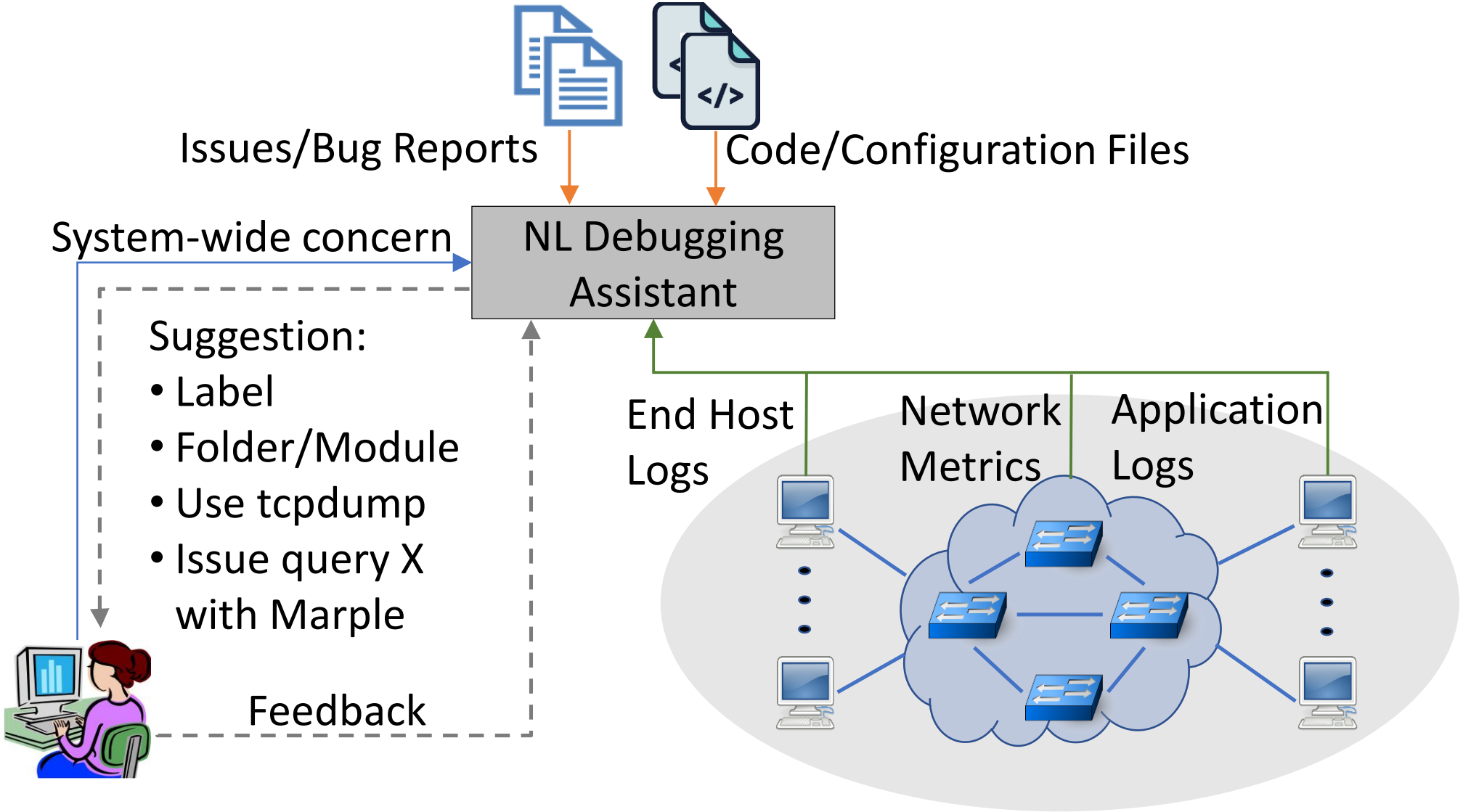
Related Work

- **Program Analysis and Synthesis:**
  - NLP for code generation, Deep API learning (FSE 16)
- **Program Debugging:**
  - Net2Text: English queries => SQL queries (NSDI 18)
- **Big Code:**
  - Initiative to perform statistical program analysis on large amounts of code

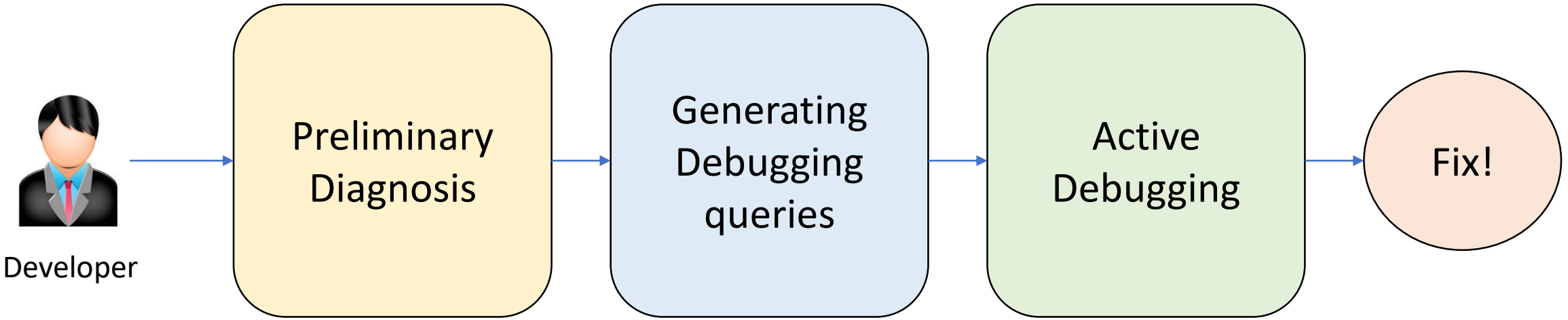
Limitations:

- Only ingest data from a single subsystem
- Assume a single-step prediction

A System-Wide Debugging Assistant Powered by Natural Language Processing

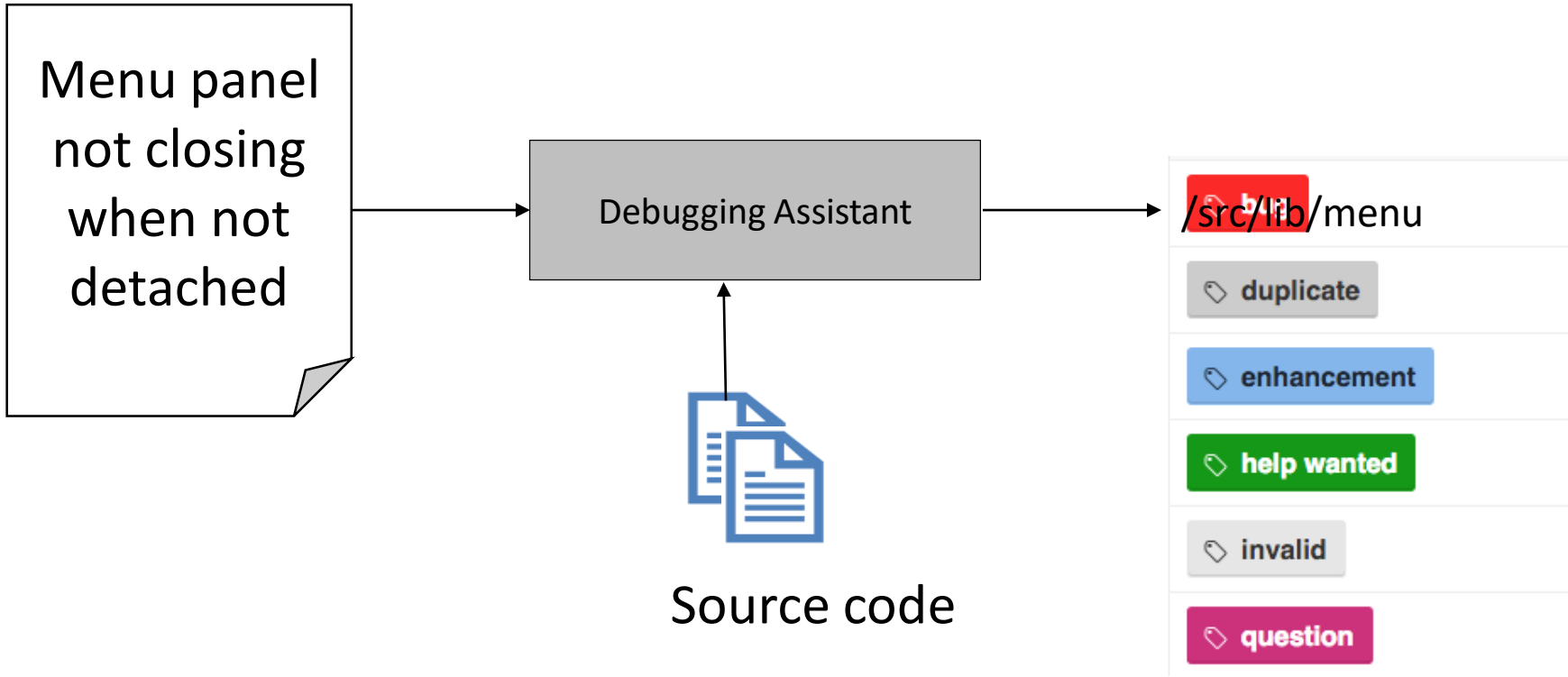


Automating steps in end-to-end debugging



Preliminary Diagnosis

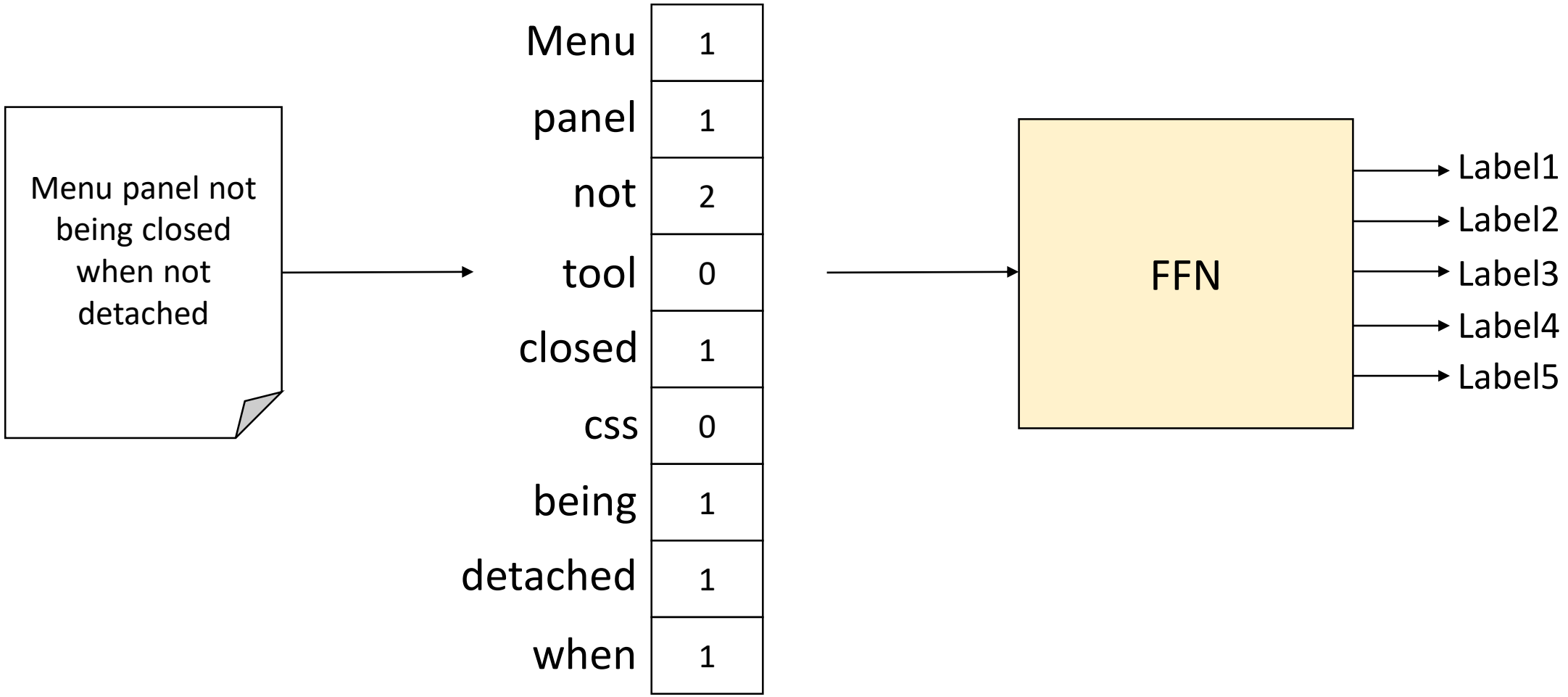
- **Automate** : Label assignment and Module prediction
- **Category** : Text classification and document retrieval



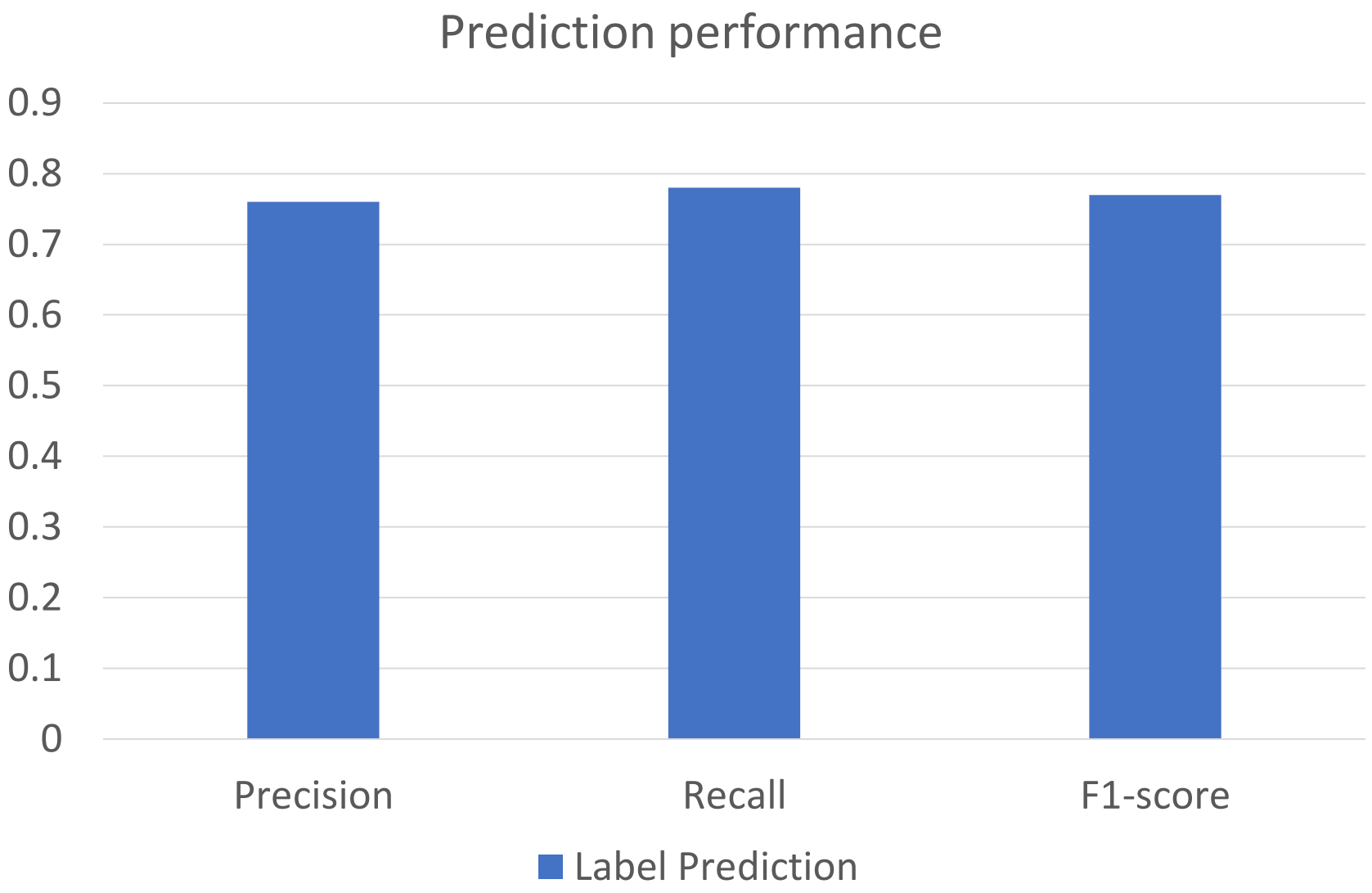
- **Challenge** : Learn joint representations of data from both unstructured text and structured source code.

Label Prediction – Preliminary Evaluation

- 165966 labeled issues from the top 98 open-source Github repositories (based on stars)
- **Bag-of-words** representation of issue text

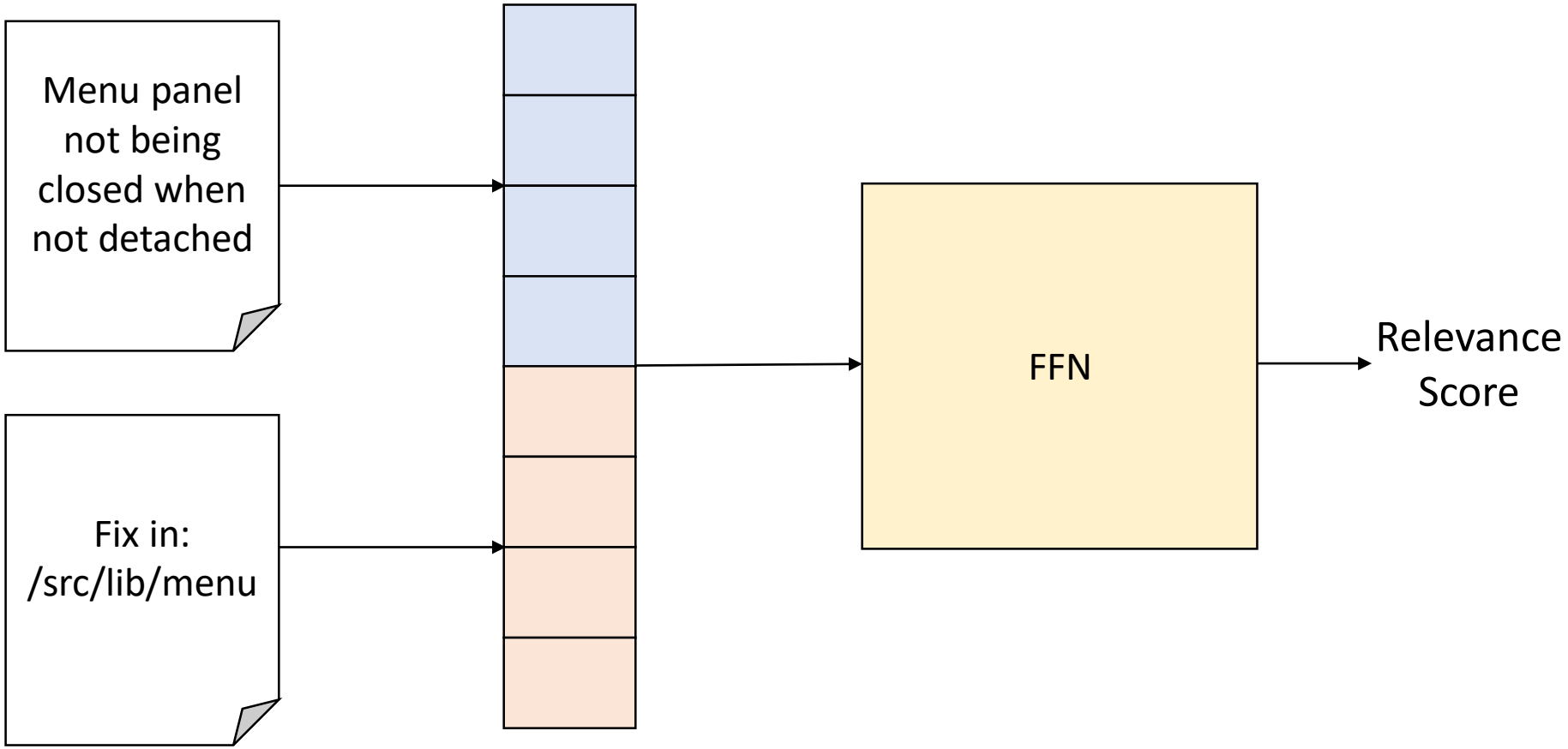


Results

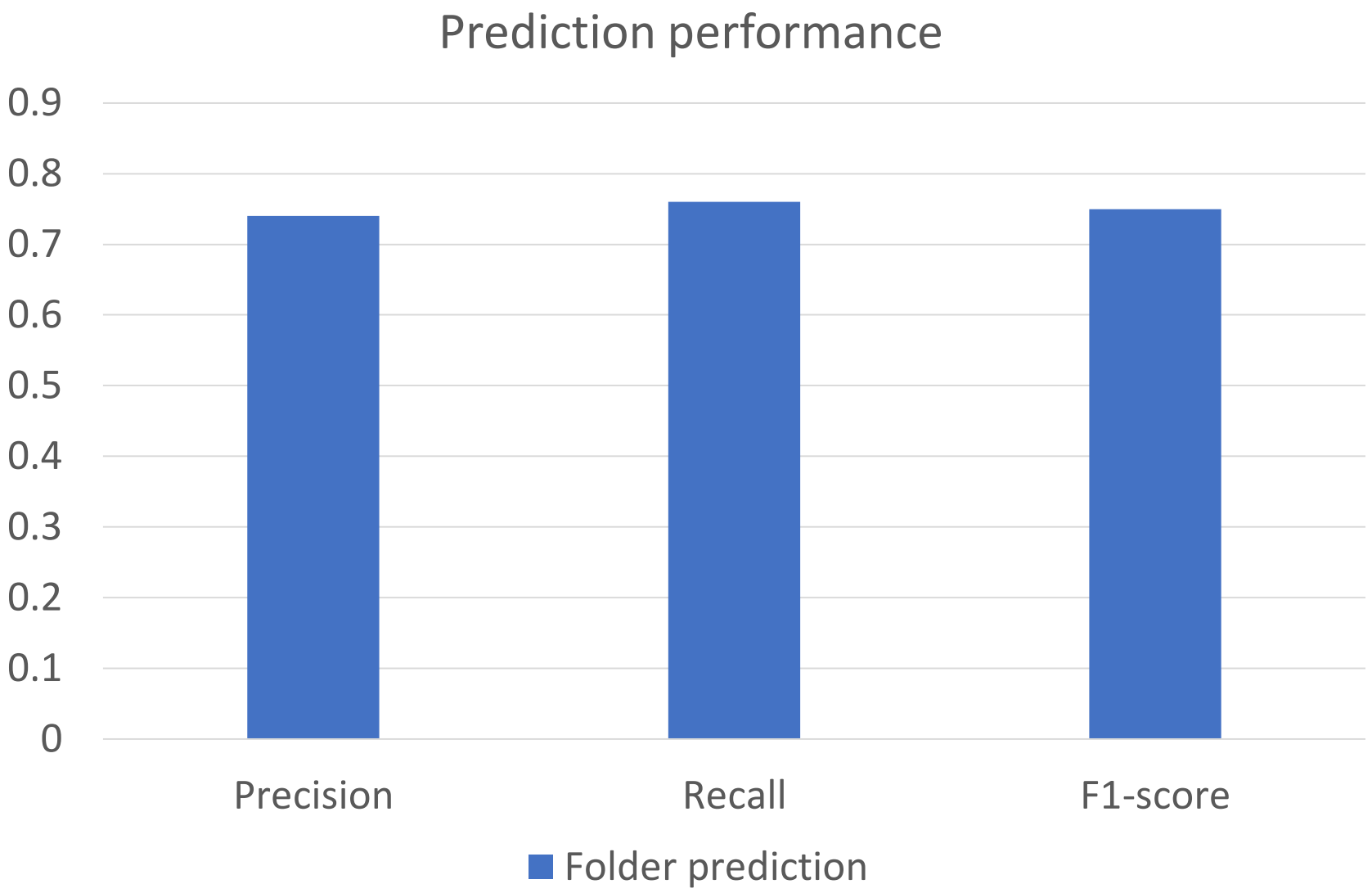

Prediction performance
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Precision
Recall
F1-score
Label Prediction

Source Code Folder Prediction – Preliminary Evaluation

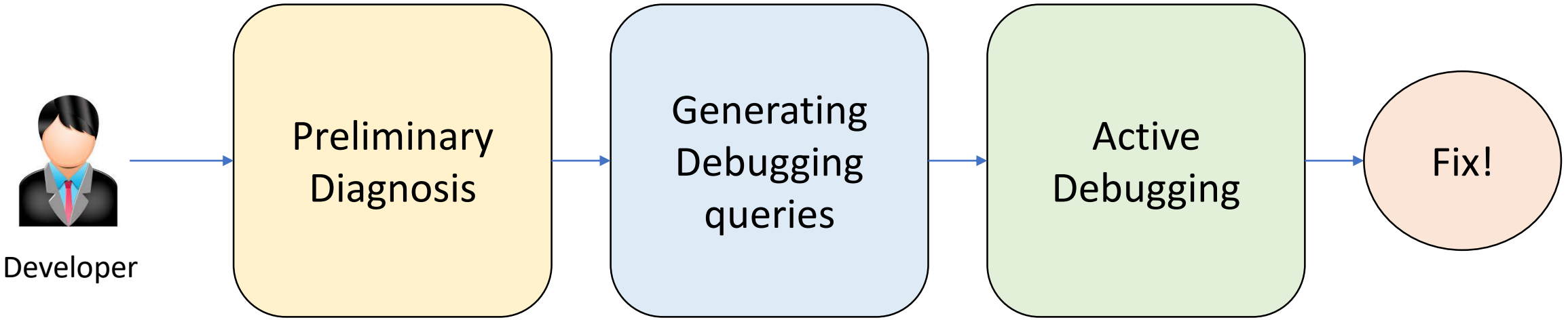
- 240138 issues with corresponding fixes from Github repositories



Results

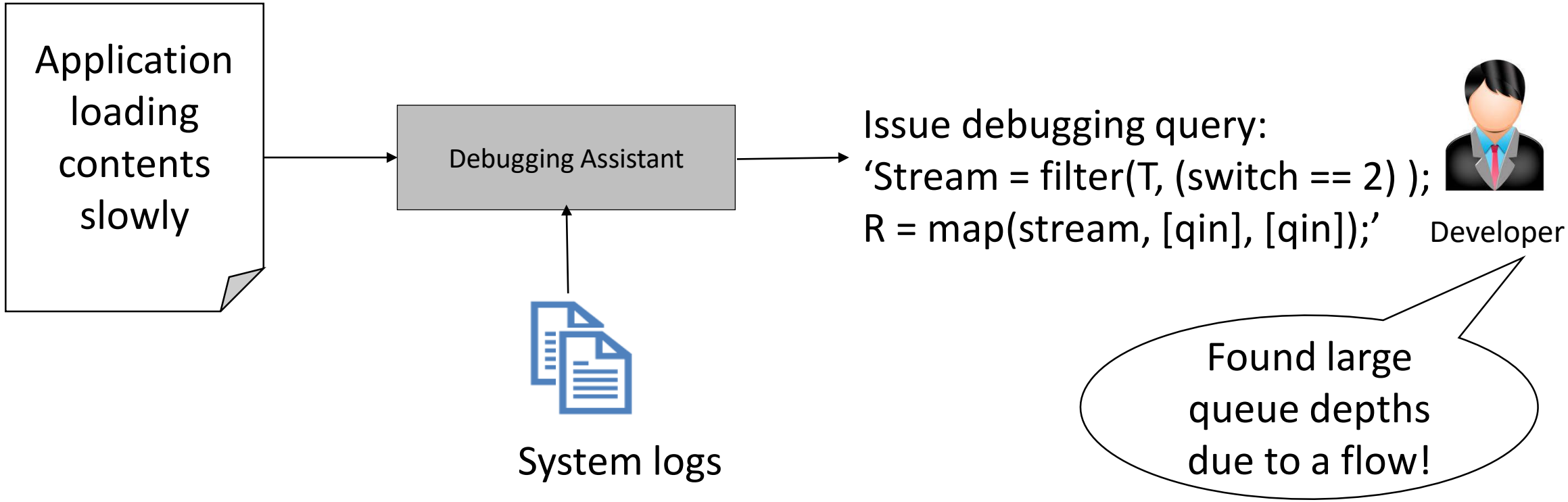

Prediction performance
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Precision
Recall
F1-score
Folder prediction

Automating steps in end-to-end debugging



Generating debugging queries

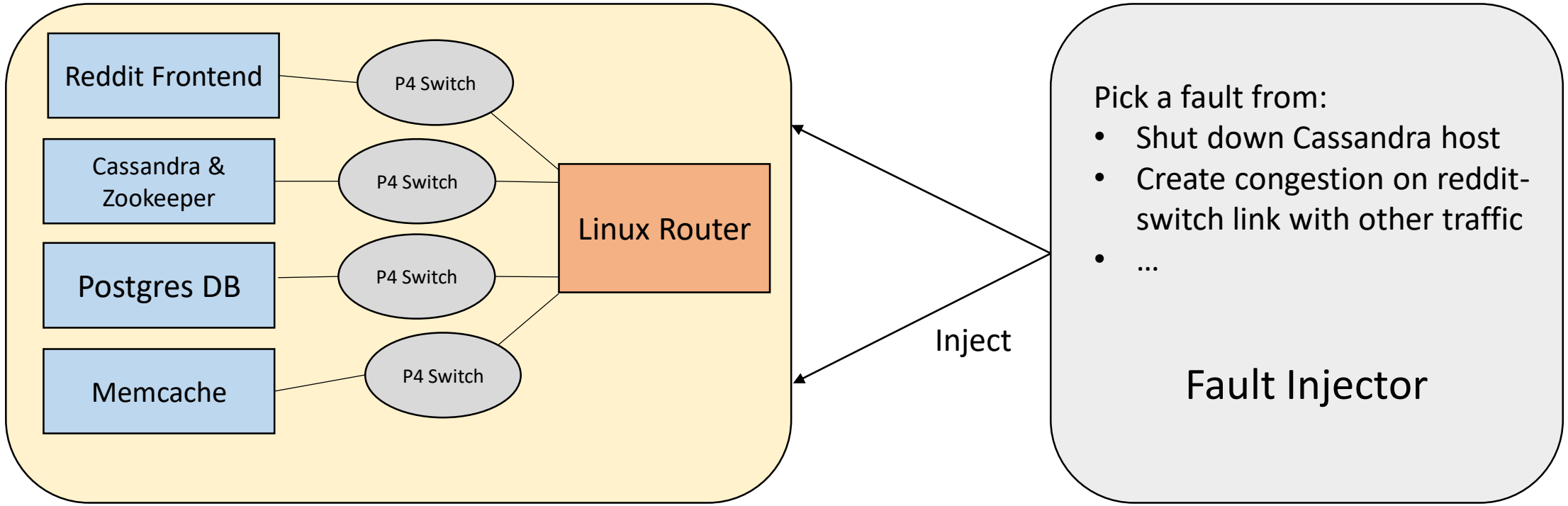
- **Automate** : Query generation for use with debugging tools
- **Category** : Language generation



- **Challenge** : Understand system logs, source code semantics and language syntax

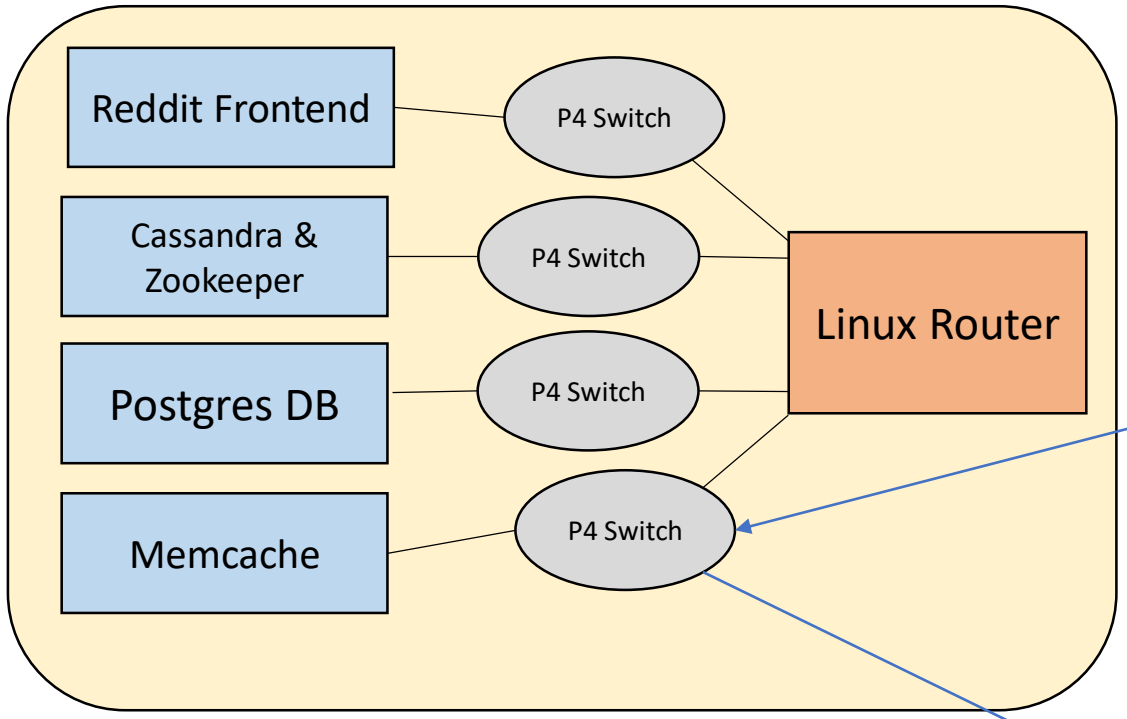
Template-based query prediction

- A platform to let users interact with the system and collect data for query generation.
- Network debugging tool for performance queries (Marple)



Distributed reddit setup

Template-based query prediction



Distributed reddit setup

P4 program

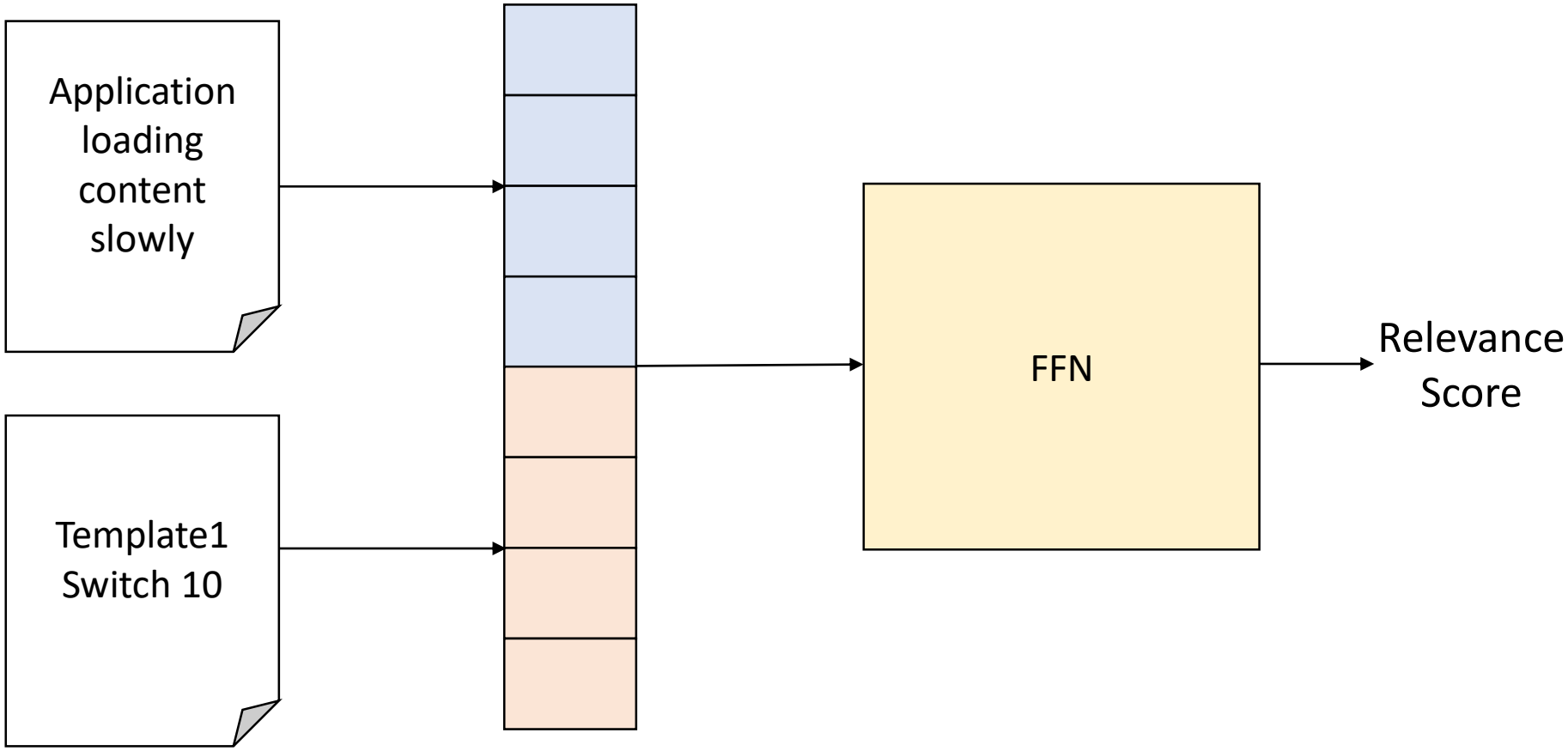
Marple

stream = filter(T, (switch == 4));
R = map(stream, [qin], [qin]);

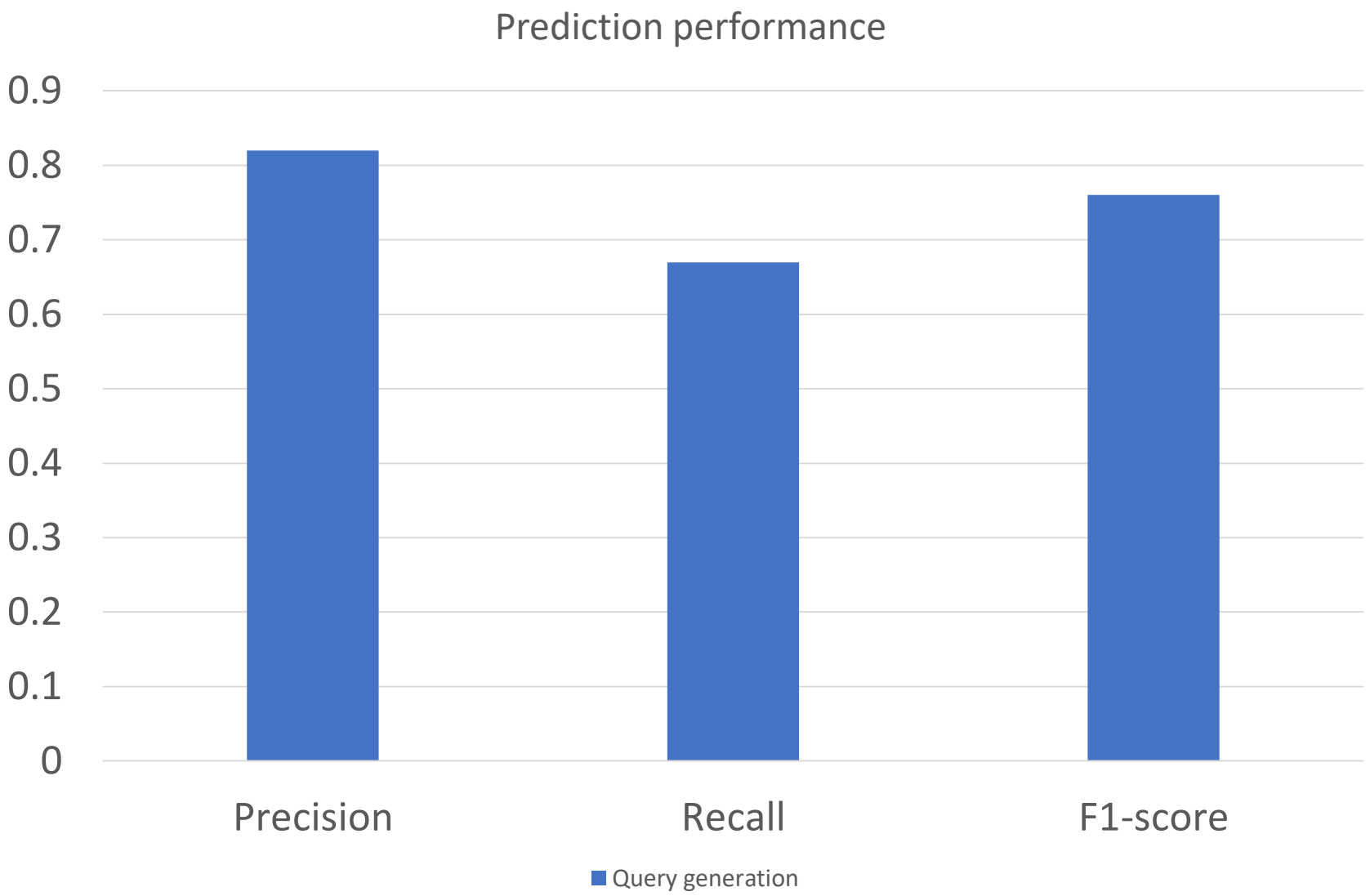
Queue depths

Template-based query prediction

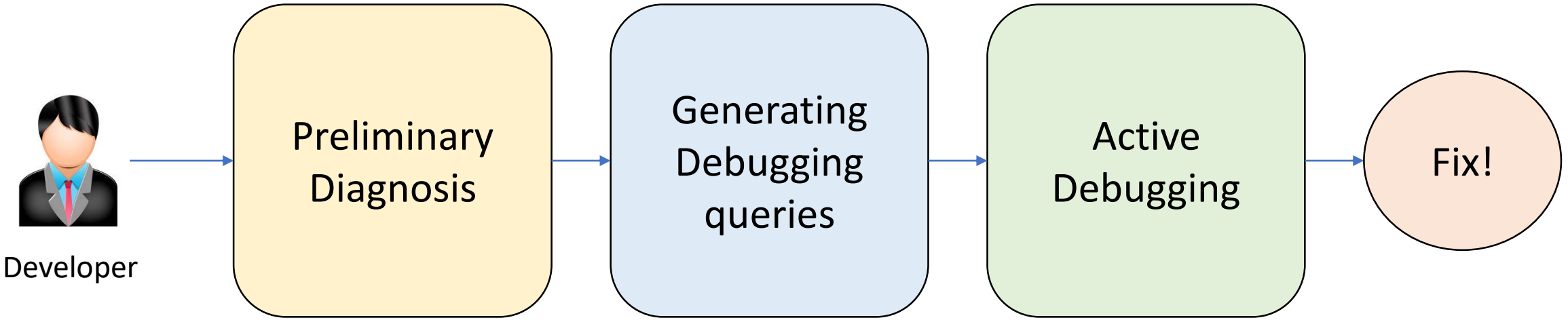
- Predict the correct template and switch to diagnose the root-cause
- Collected issue reports using the testbed from one user for faults injected using fault injector.



Results

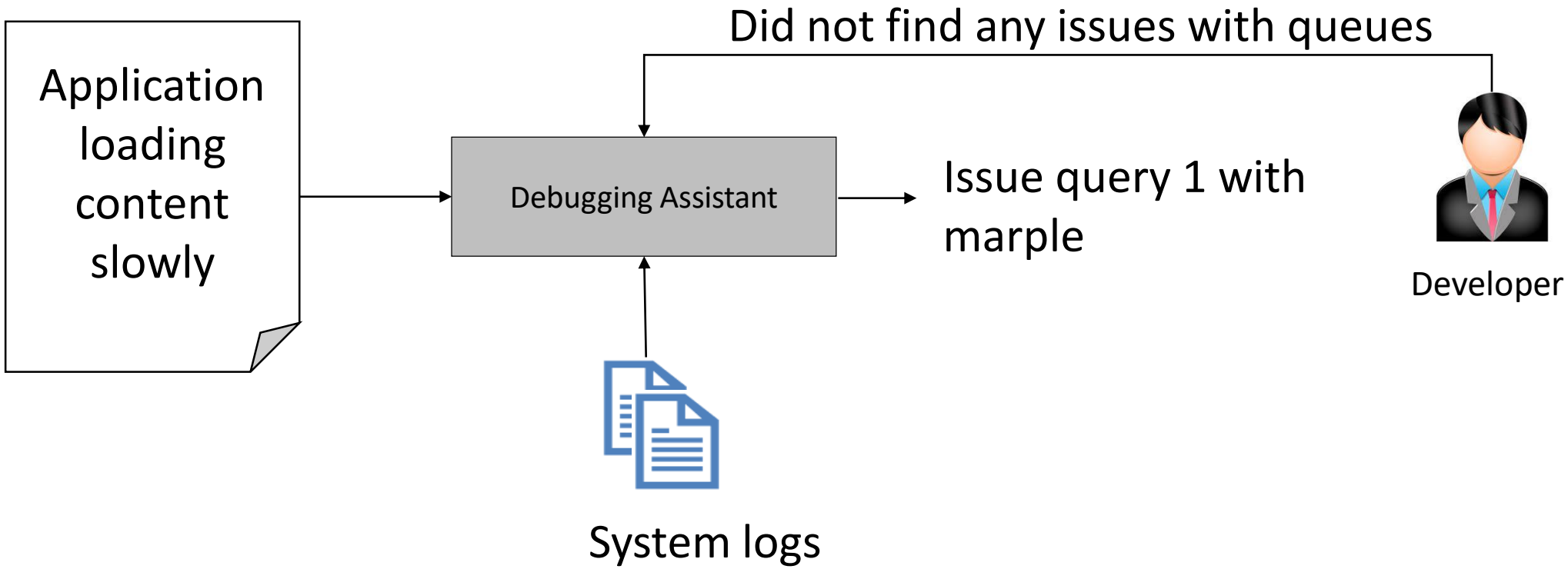

Prediction performance
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Precision
Recall
F1-score
Query generation

Automating steps in end-to-end debugging



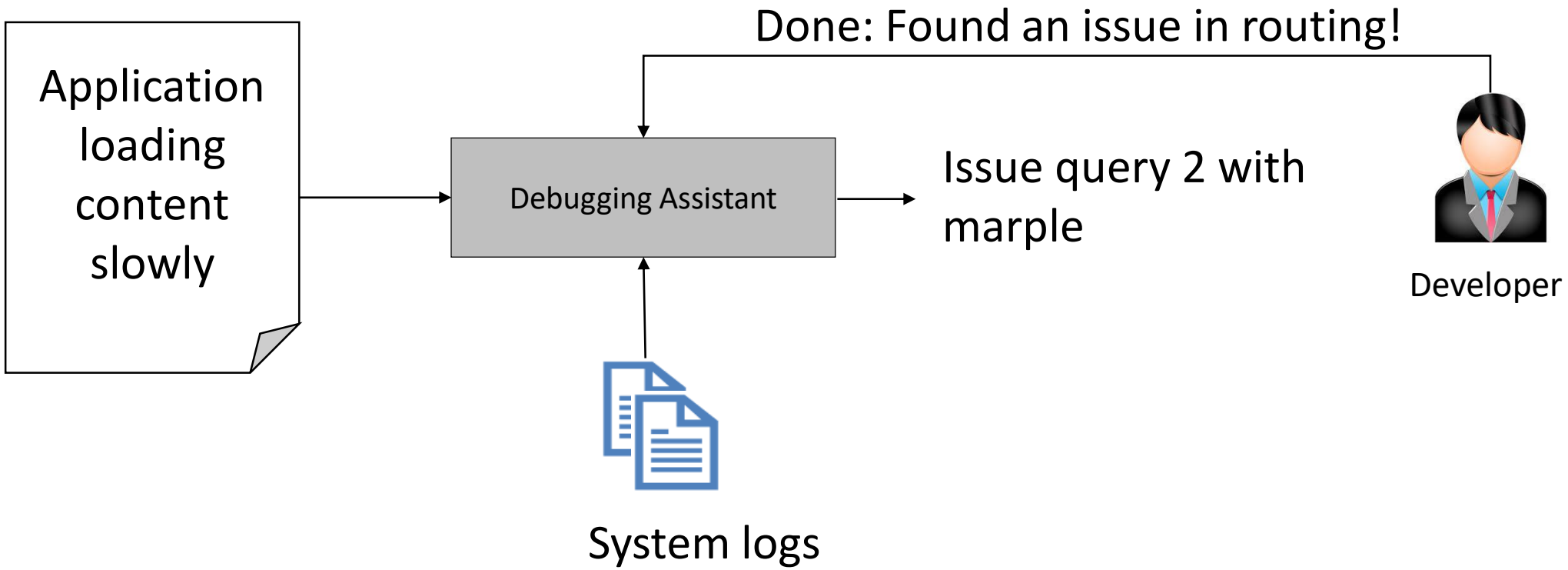
Active (interactive) debugging

- **Automate** : Iterative query generation by incorporating feedback
- **Category** : Sequential decision making



Active (interactive) debugging

- **Automate** : Iterative query generation by incorporating feedback
- **Category** : Sequential decision making



- **Challenge** : Developer-assistant interface to leverage developer's experience

Challenges & Future Work

- Need to determine optimal model to leverage information from text and traces to generate queries syntactically
- Data collection, training time – need to develop novel systems and algorithmic techniques
- End-to-end evaluation – Evaluate impact of the assistant in the debugging experience with real issues.
- Developer study on systems with reasonable complexity

Conclusion

- Our work paints a vision for an **end-to-end** debugging assistant which can:
  - Process natural language inputs
  - Various system logs
  - Leverage multiple domain specific debugging tools
  - Automate the three steps in debugging

Thank you!

Contact: dogga@cs.ucla.edu
http://web.cs.ucla.edu/~dogga