

# Refining Network Intents for Self-Driving Networks

---

Arthur Selle Jacobs<sup>1</sup>

Ricardo José Pfitscher<sup>1</sup>, Ronaldo Alves Ferreira<sup>2</sup>, Lisandro Zambenedetti Granville<sup>1</sup>

<sup>1</sup>UFRGS

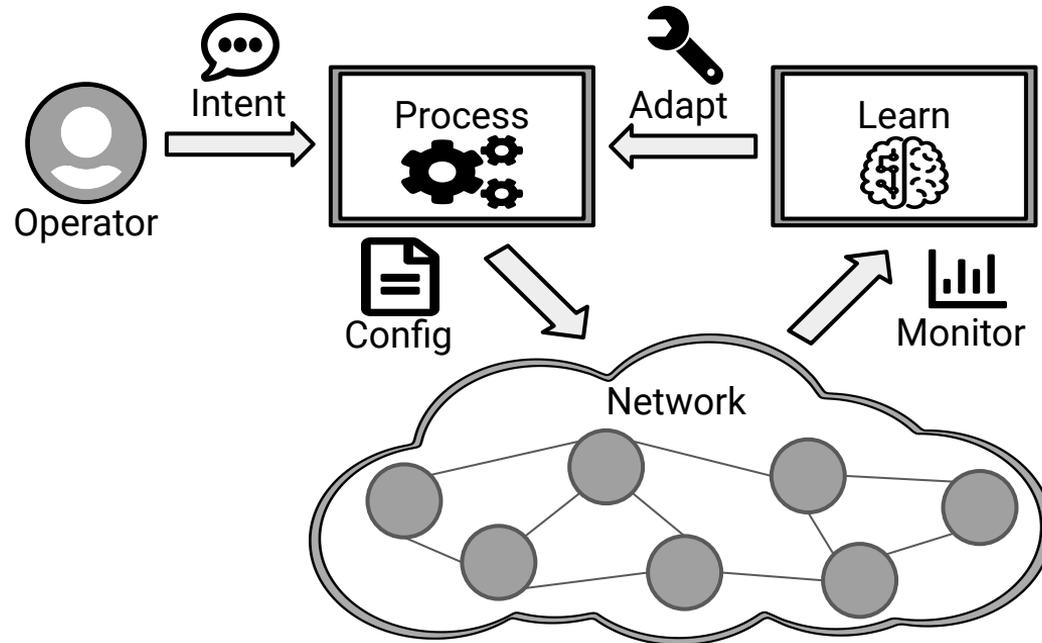
<sup>2</sup>UFMS

Montreal, Canada  
July 25, 2019



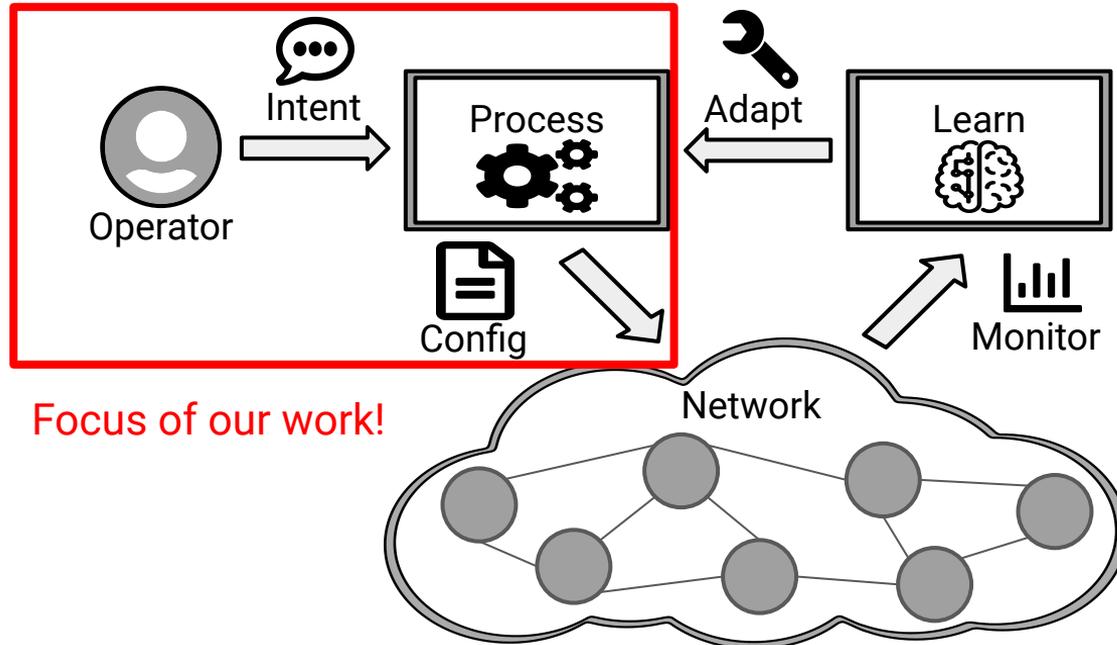
# Self-Driving Networks

## High-level Architecture



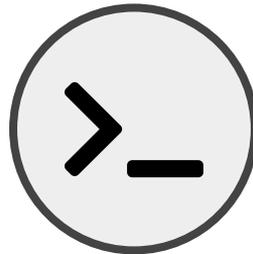
# Self-Driving Networks

## High-level Architecture



Focus of our work!

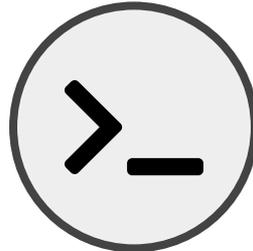
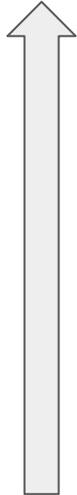
# Nowadays...



IETF 105, NMRG 54th Meeting

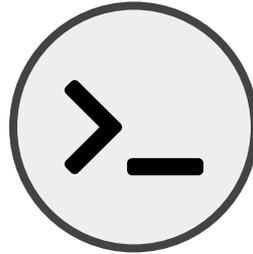
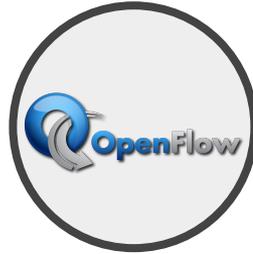
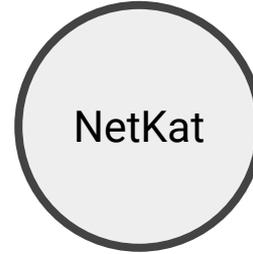
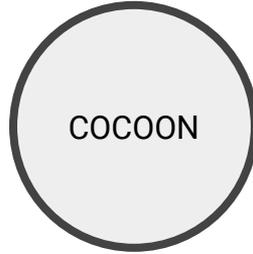
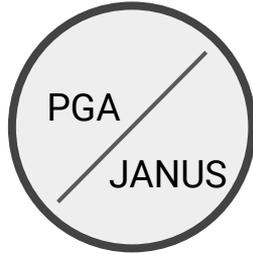
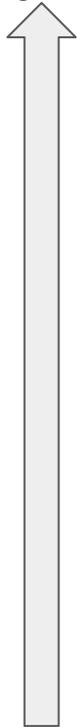
# Nowadays...

Higher-level



# Nowadays...

Higher-level



How to deploy intents expressed in natural language?

# Network Intent Refinement using *Nile*

# Network Intent Refinement using *Nile*

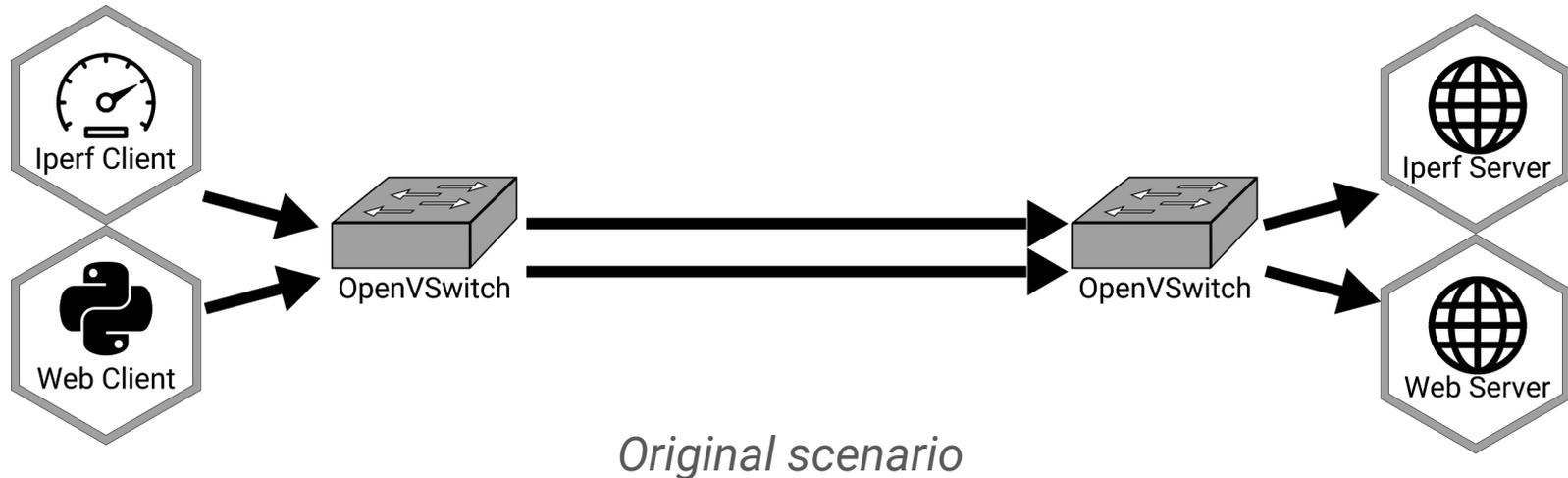
1. Receive intents expressed in natural language

# Network Intent Refinement using *Nile*

1. Receive intents expressed in natural language
2. Use *Nile* to ask for operator feedback

# Intent Refinement By Example

Experimental Service Chaining scenario, using SONATA-NFV and Mininet



# Intent Refinement By Example

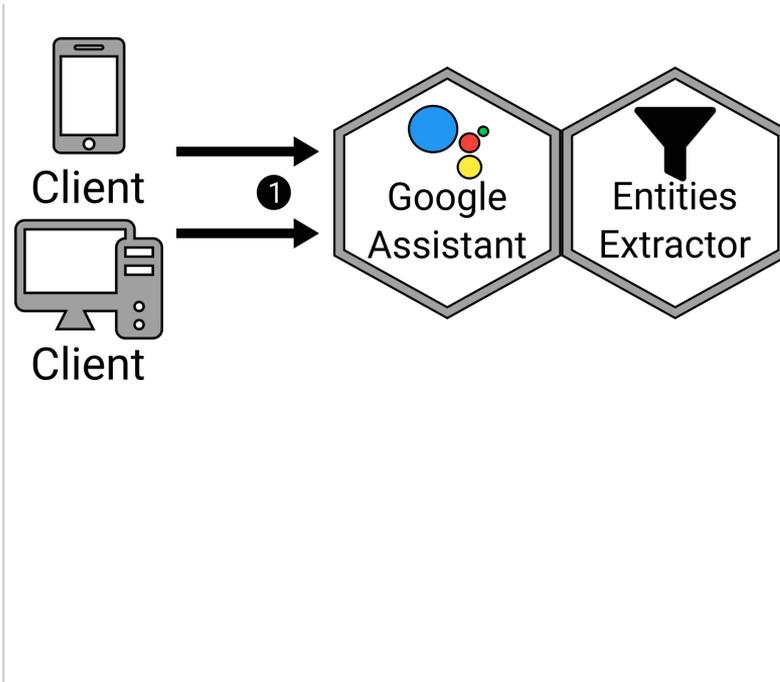
“Please add a firewall and an IDS  
from Iperf client to server”

*Original Intent*

# Intent Refinement By Example

“Please add a firewall and an IDS  
from Iperf client to server”

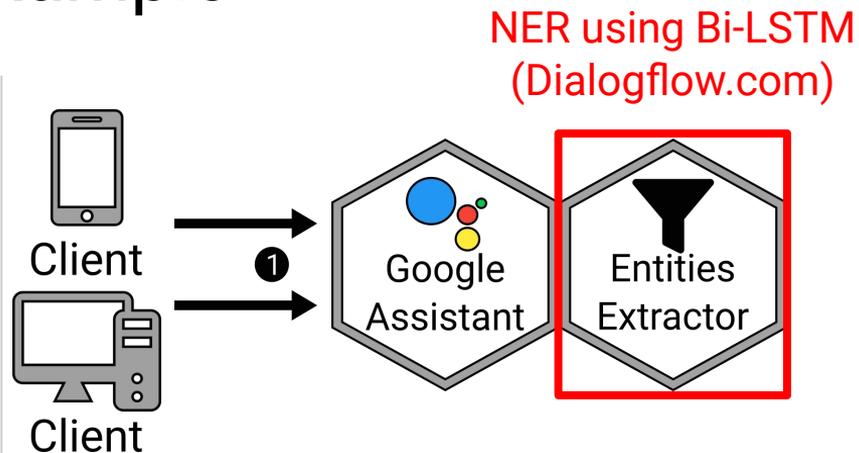
*Original Intent*



# Intent Refinement By Example

“Please add a firewall and an IDS  
from Iperf client to server”

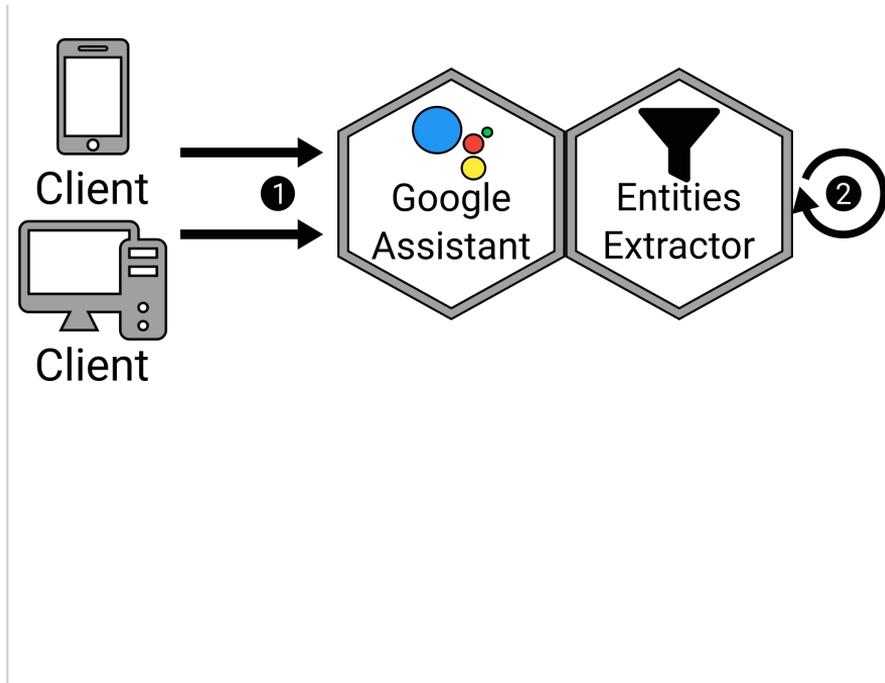
*Original Intent*



# Intent Refinement By Example

“Please add a **firewall** and an **IDS**  
from **Iperf client** to **server**”

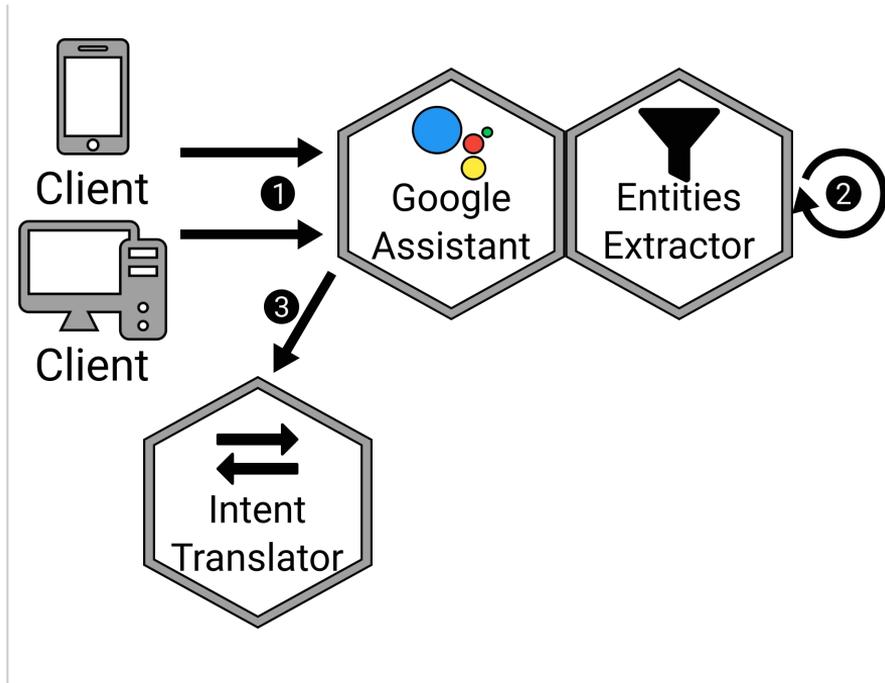
*Extracted entities*



# Intent Refinement By Example

“Please add a **firewall** and an **IDS**  
from **Iperf client** to **server**”

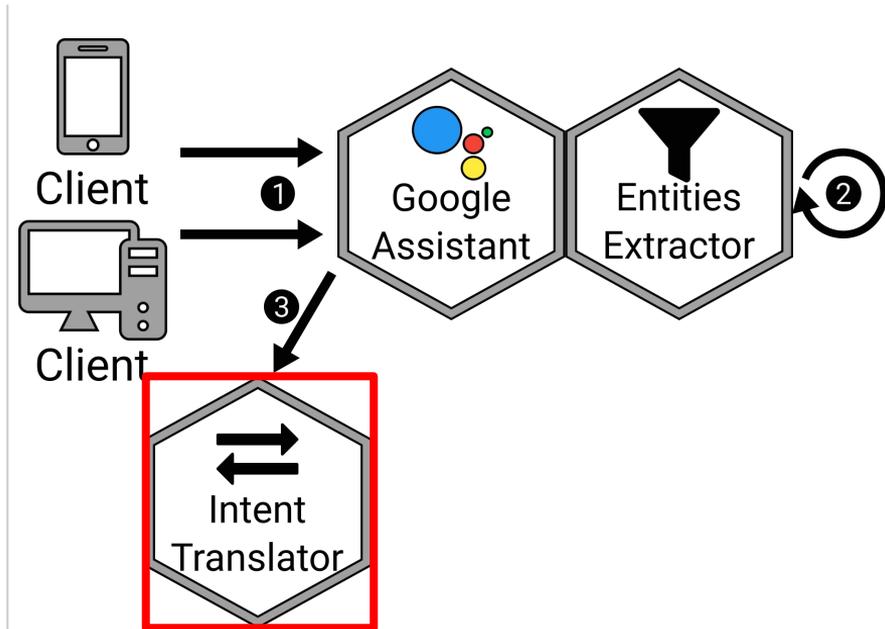
*Extracted entities*



# Intent Refinement By Example

“Please add a **firewall** and an **IDS**  
from **Iperf client** to **server**”

*Extracted entities*

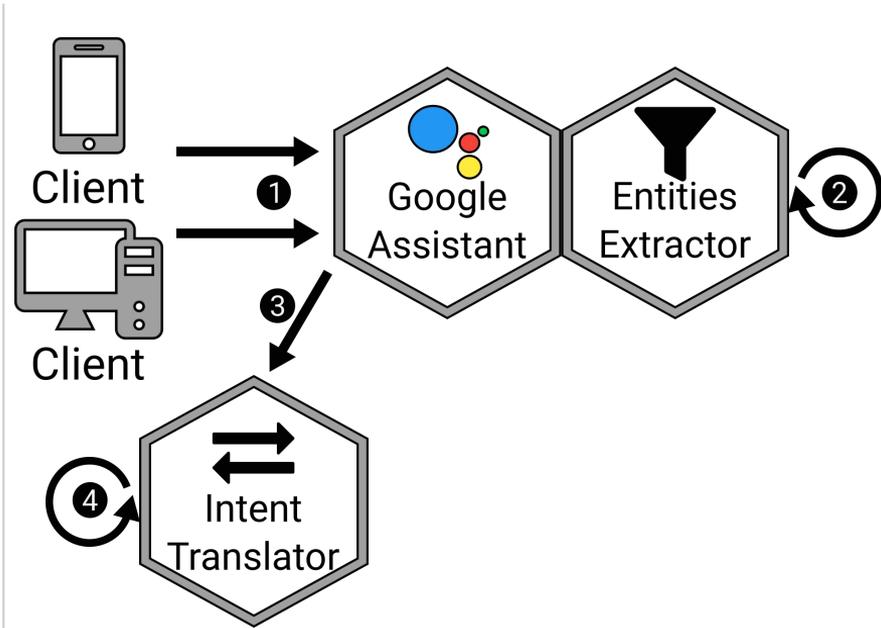


Neural Sequence to Sequence learning model,  
using Recursive Neural Networks.

# Intent Refinement By Example

```
define intent testIntent:  
  from endpoint('iperf client')  
  to endpoint('iperf server')  
  add middlebox('firewall'),  
  middlebox('ids')
```

*Nile intent*



# Intent Refinement By Example

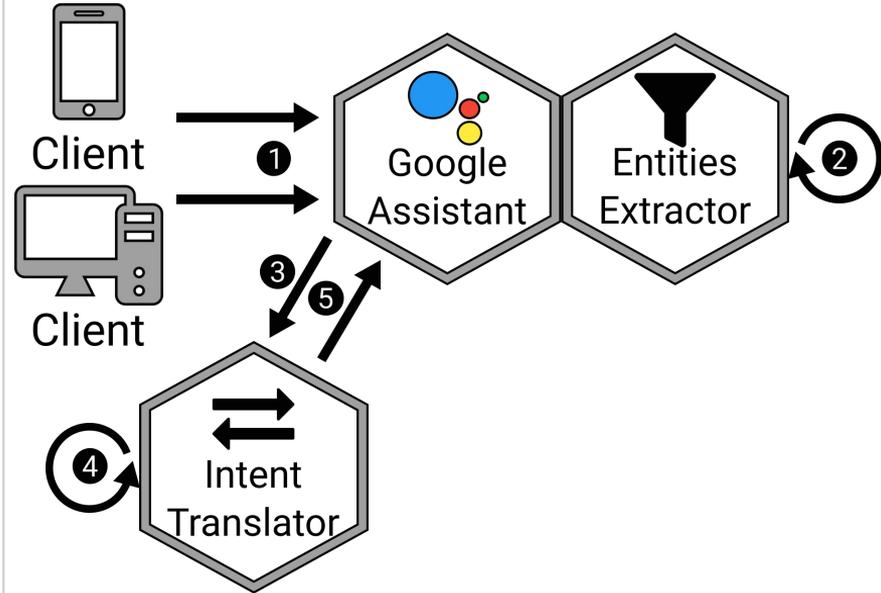
Is this what you want?

```
define intent testIntent:  
  from endpoint('iperf client')  
  to endpoint('iperf server')  
  add middlebox('firewall'),  
  middlebox('ids')
```

*Nile intent*

YES

NO



# Intent Refinement By Example

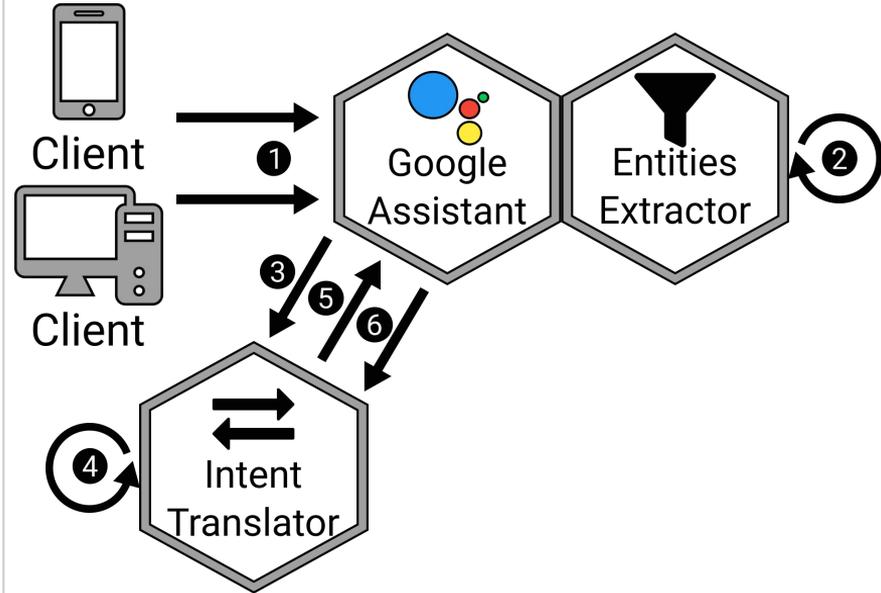
Is this what you want?

```
define intent testIntent:  
  from endpoint('iperf client')  
  to endpoint('iperf server')  
  add middlebox('firewall'),  
  middlebox('ids')
```

*Nile intent*

YES

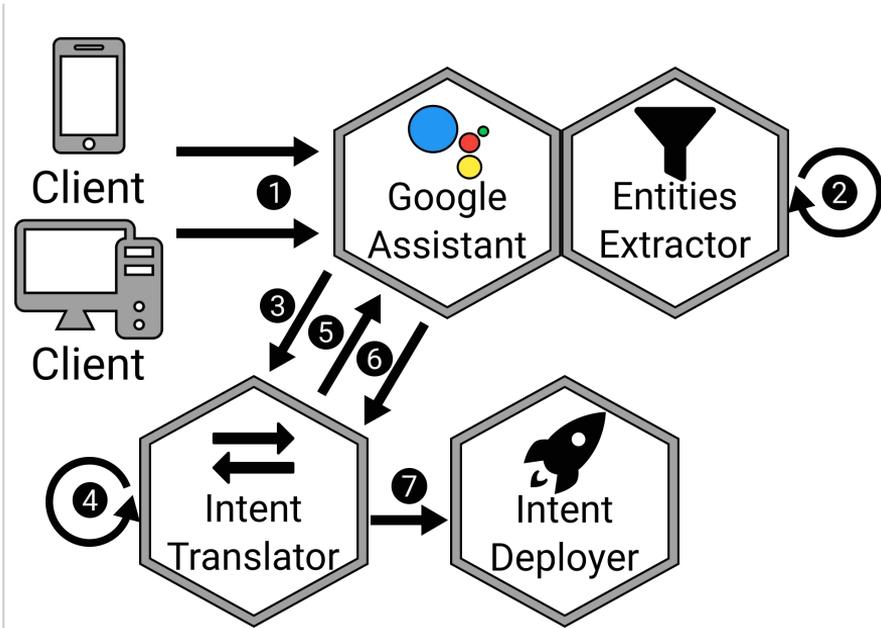
NO



# Intent Refinement By Example

```
define intent testIntent:  
  from endpoint('iperf client')  
  to endpoint('iperf server')  
  add middlebox('firewall'),  
  middlebox('ids')
```

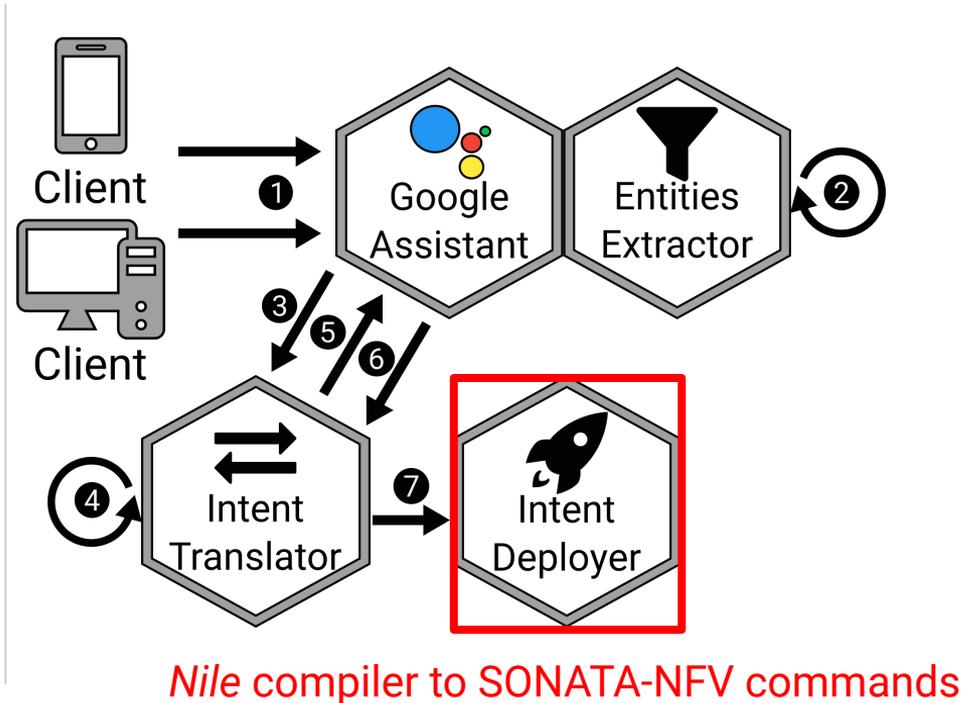
*Nile intent*



# Intent Refinement By Example

```
define intent testIntent:  
  from endpoint('iperf client')  
  to endpoint('iperf server')  
  add middlebox('firewall'),  
  middlebox('ids')
```

*Nile intent*

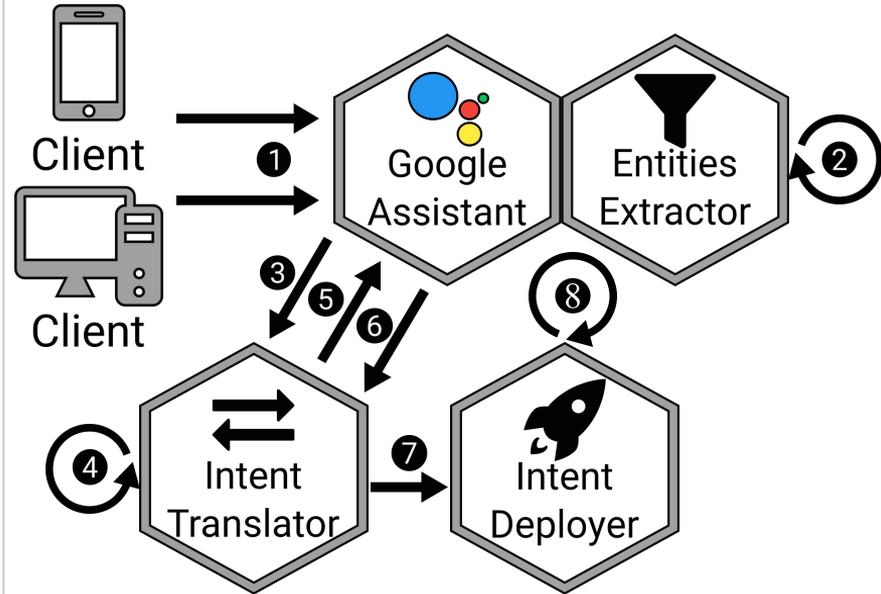


# Intent Refinement By Example

```
# deploy vnfs
vim-emu compute start -n fw <params>
vim-emu compute start -n ids <params>

# chain vnfs
vim-emu network add -b -src
iperf-c:c-eth0 -dst fw:in
vim-emu network add -b -src fw:out -dst
ids:in
vim-emu network add -b -src ids:out -dst
iperf-s:s-eth0
```

*Compiled SONATA-NFV commands*

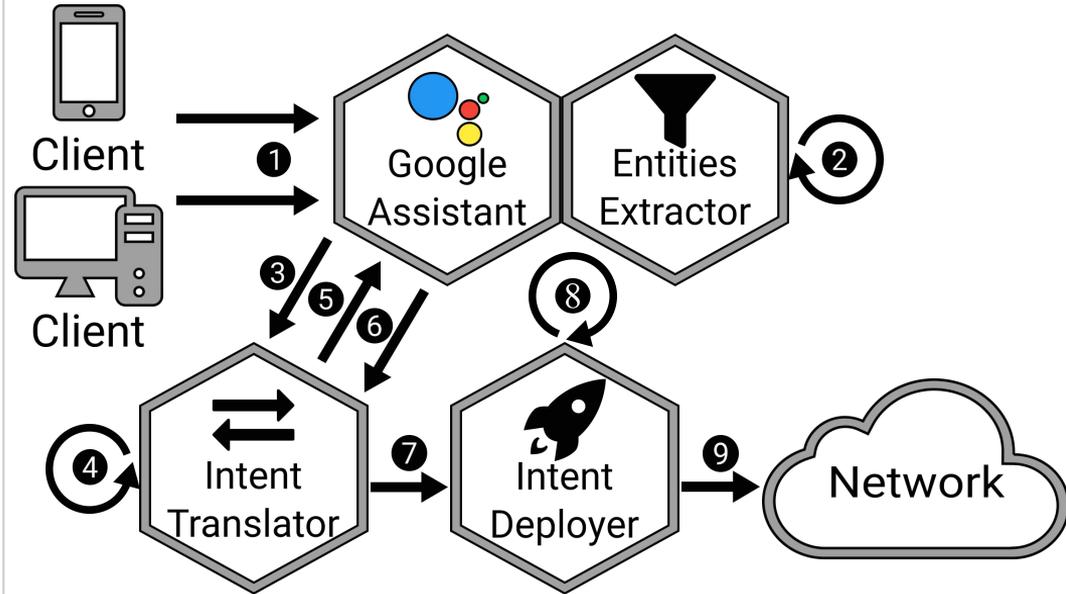


# Intent Refinement By Example

```
# deploy vnfs
vim-emu compute start -n fw <params>
vim-emu compute start -n ids <params>

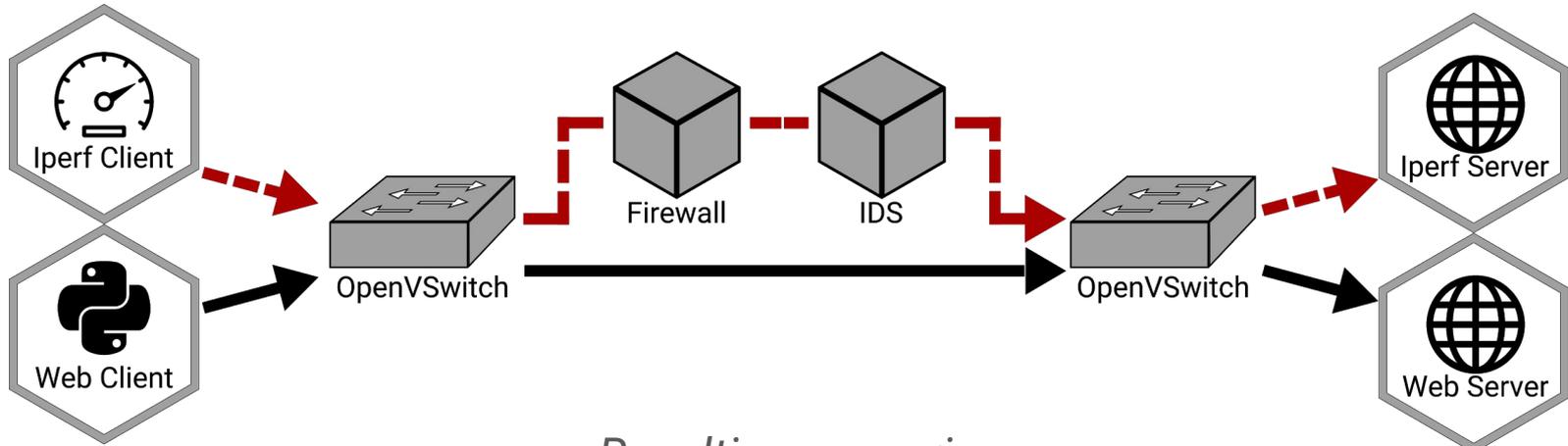
# chain vnfs
vim-emu network add -b -src
iperf-c:c-eth0 -dst fw:in
vim-emu network add -b -src fw:out -dst
ids:in
vim-emu network add -b -src ids:out -dst
iperf-s:s-eth0
```

*Compiled SONATA-NFV commands*



# Intent Refinement By Example

“Please add a **firewall** and an **IDS** from **Iperf client** to **server**”



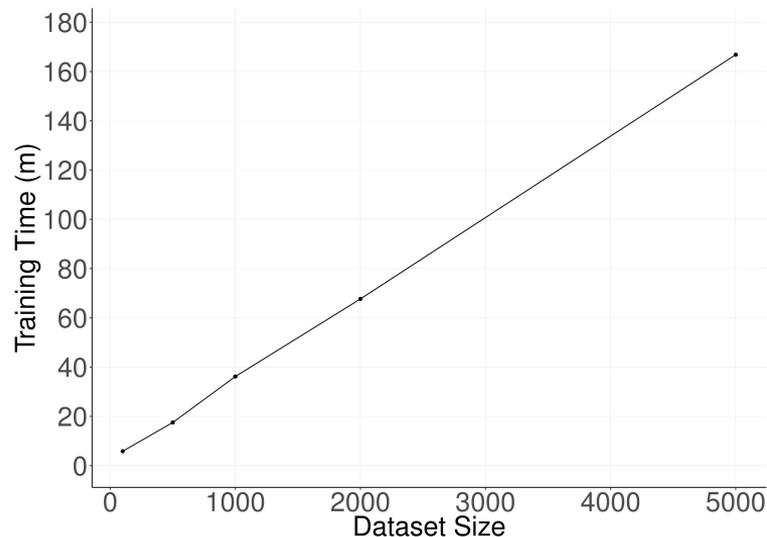
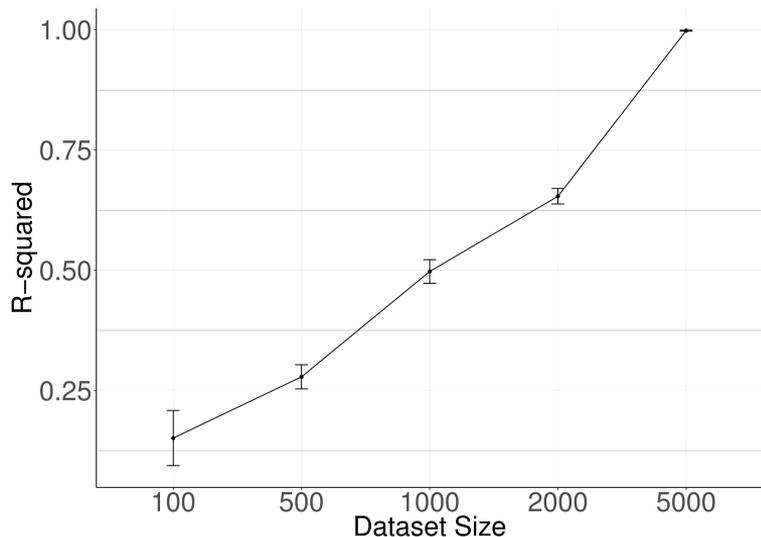
*Resulting scenario*

# Evaluation

- (i) The accuracy we can achieve with different sizes of training datasets, aiming to find the optimal ratio between dataset size and prediction accuracy.
- (ii) The impact of the operator feedback on the accuracy of predictions over time to determine if it improves accuracy.
  - 5 dataset sizes:
    - 100, 500, 1000, 2000, 5000 entries.
    - 20% validation split.
  - We generated the datasets automatically with random sets of *entities* and *Nile* intent pairs, combining a different number of middleboxes, endpoints, traffic matching rules, time, and QoS requirements in each intent.

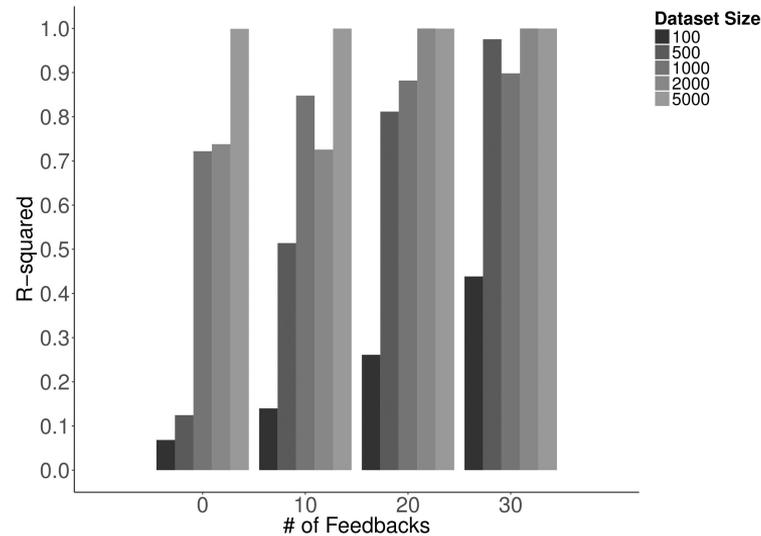
# Results

(i) The accuracy we can achieve with different sizes of training datasets, aiming to find the optimal ratio between dataset size and prediction accuracy.



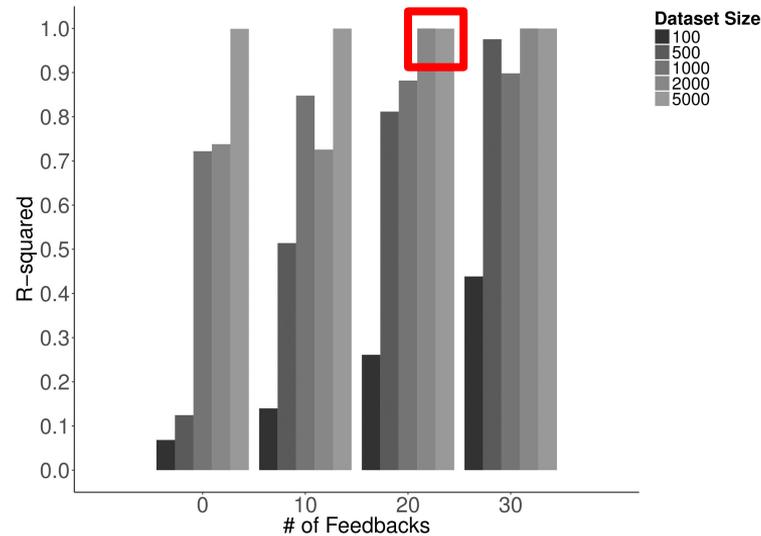
# Results

(ii) The impact of the operator feedback on the accuracy of predictions over time to determine if it improves accuracy.



# Results

(ii) The impact of the operator feedback on the accuracy of predictions over time to determine if it improves accuracy.



# Summary

“How to deploy network intents expressed as natural language?”

**Using our refinement process + *Nile***

Low-level of technical knowledge required

Feedback from user allows to learn over time

“What’s next?”

Fully implement *Nile* compilation into OpenFlow and P4 backends.

Further evaluate the end-to-end proposed solution.

# Thank you!

Arthur Jacobs  
[asjacobs@inf.ufrgs.br](mailto:asjacobs@inf.ufrgs.br)

