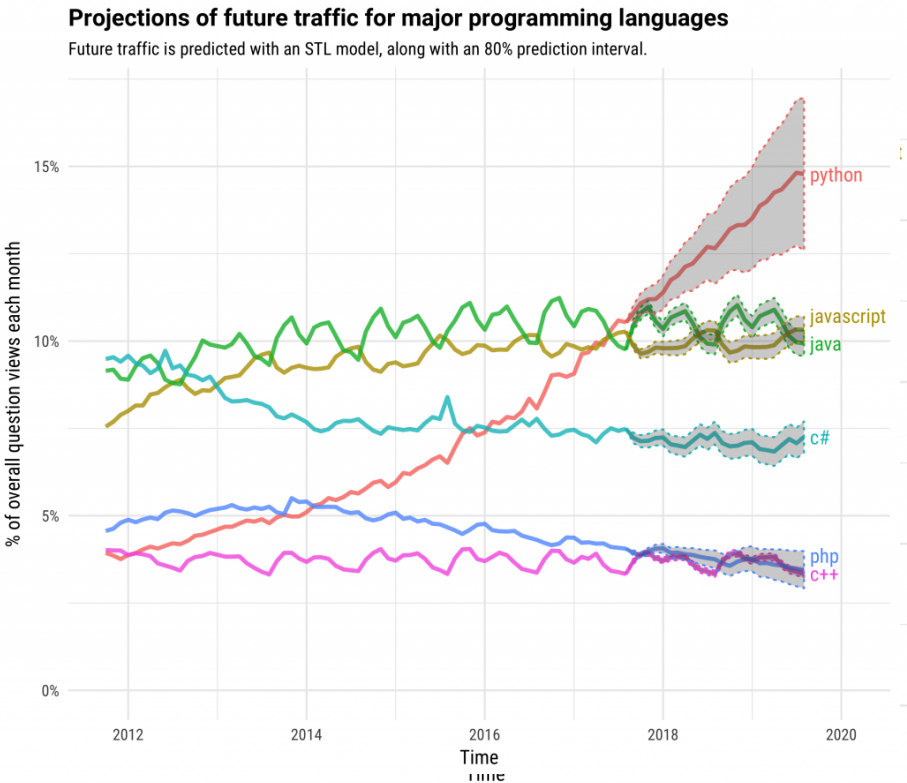
Nagini: A Static Verifier for Python

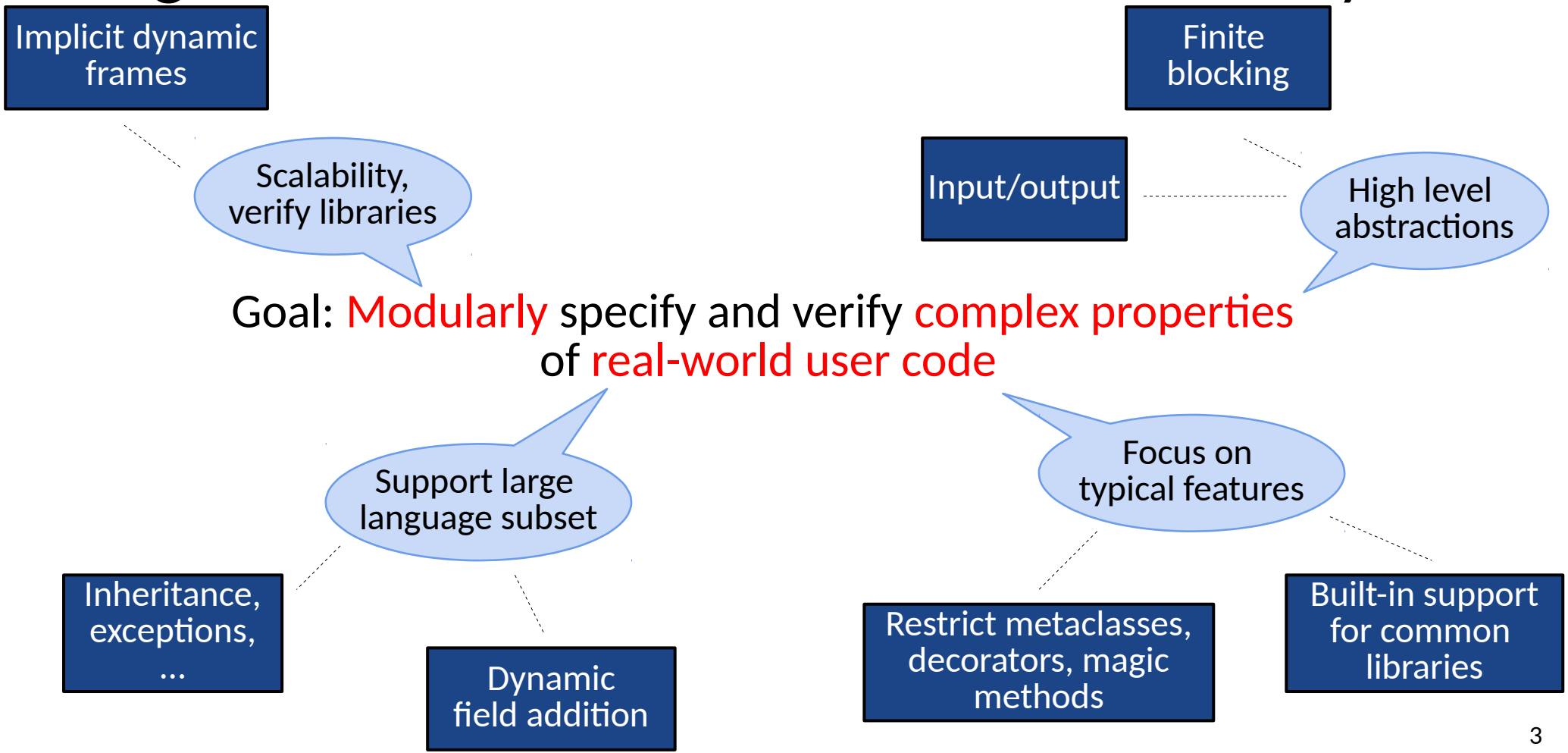
Marco Eilers
Peter Müller

ETH zürich

python™


Projections of future traffic for major programming languages
Future traffic is predicted with an STL model, along with an 80% prediction interval.
% of overall question views each month
15%
10%
5%
0%
2012
2014
2016
2018
2020
Time
python
javascript
java
c#
php
c++

Nagini: Sound Verification for Concurrent Python



Nagini: Sound Verification for Concurrent Python

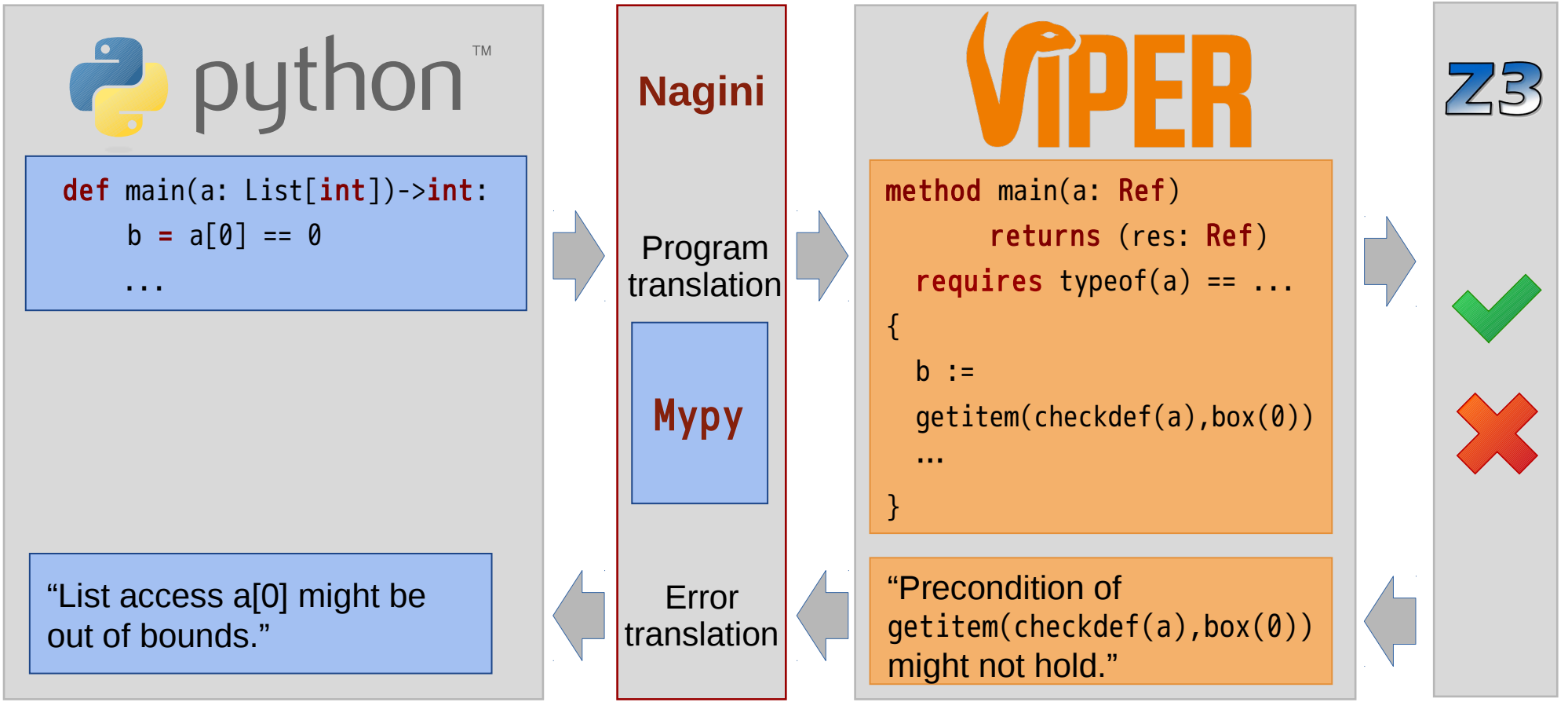
```
from nagini_contracts.contracts import *

def indexOf(l: List[int], i: int, val: int) -> int:
    Requires(i >= 0 and i < len(l))
    Requires(list_pred(l))
    Requires(MustTerminate(len(l) - i))
    Ensures(list_pred(l))
    Ensures(Implies(Result() != -1,
            Forall(range(i, Result()), lambda j: l[j] != val) and
            l[Result()] == val))
    if l[i] == val:
        return i
    if i >= len(l) - 1:
        return -1
    return indexOf(l, i + 1, val)
```

- Memory Safety
  - No runtime errors
  - No data races
  - …
- Functional properties
- Termination

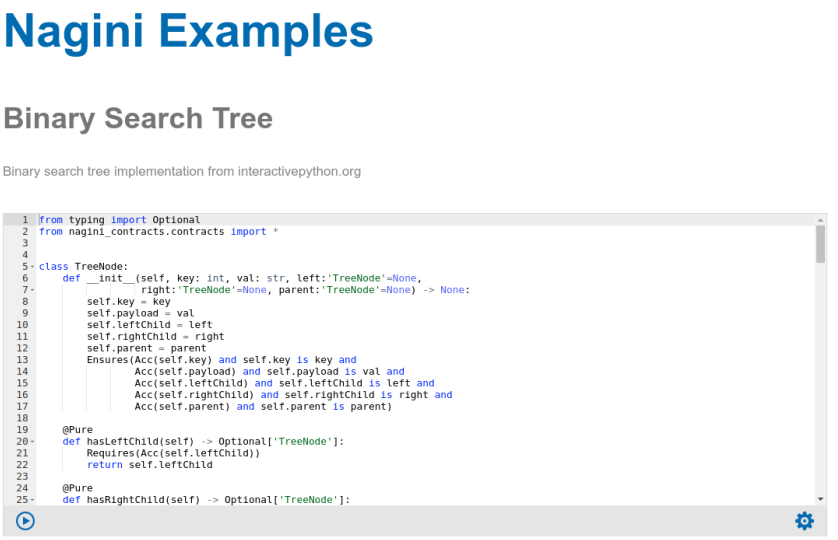
- Finite blocking
  - Deadlock freedom
  - …
- Input/Output behavior
  - Allowed I/O
  - Liveness

Implementation



Conclusion

- In the paper: Experimental evaluation
- Available open source
  - www.github.com/marcoeilers/nagini
  - Online version available:
    viper.ethz.ch/nagini-examples



- Currently used to verify SCION internet architecture implementation
- Future work: Higher order functions, secure information flow

Try it out!

www.github.com/marcoeilers/nagini

viper.ethz.ch/nagini-examples/