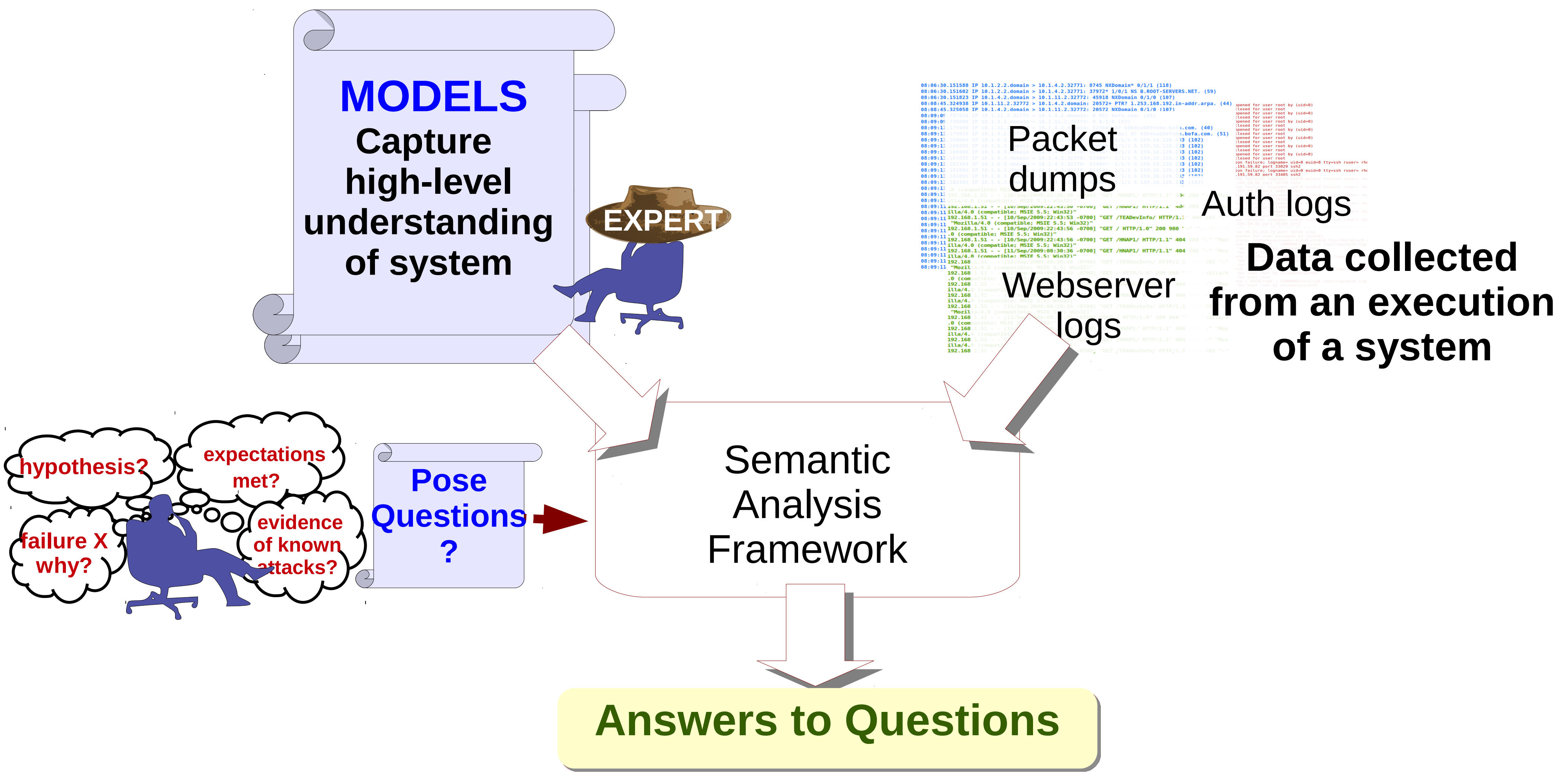
A Semantic Framework for Data Analysis in Networked Systems

Arun Viswanathan, Alefiya Hussain, Jelena Mirkovic, Stephen Schwab, John Wroclawski

USC/Information Sciences Institute

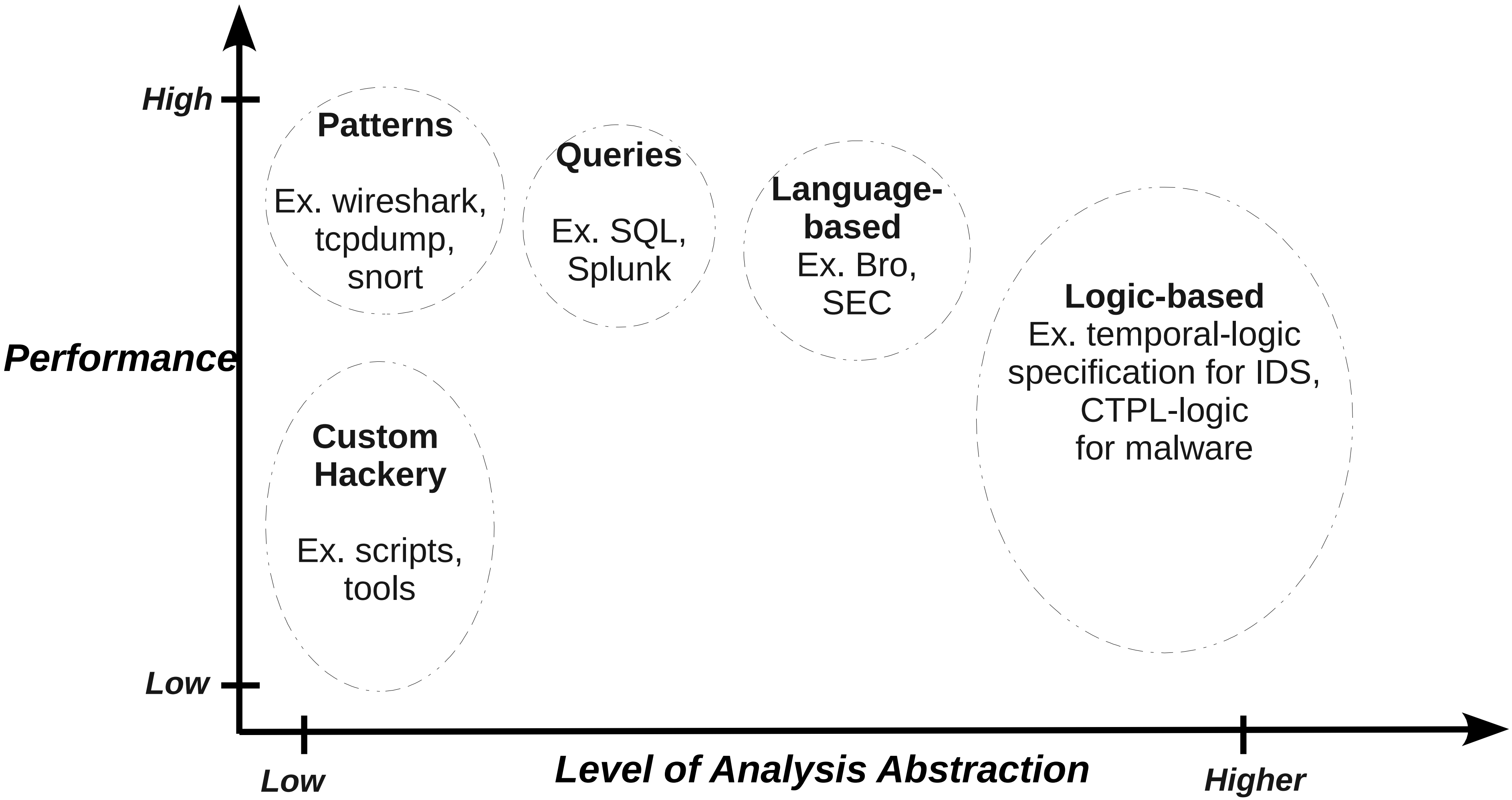
Data Analysis in Networked Systems

Our Semantic Approach

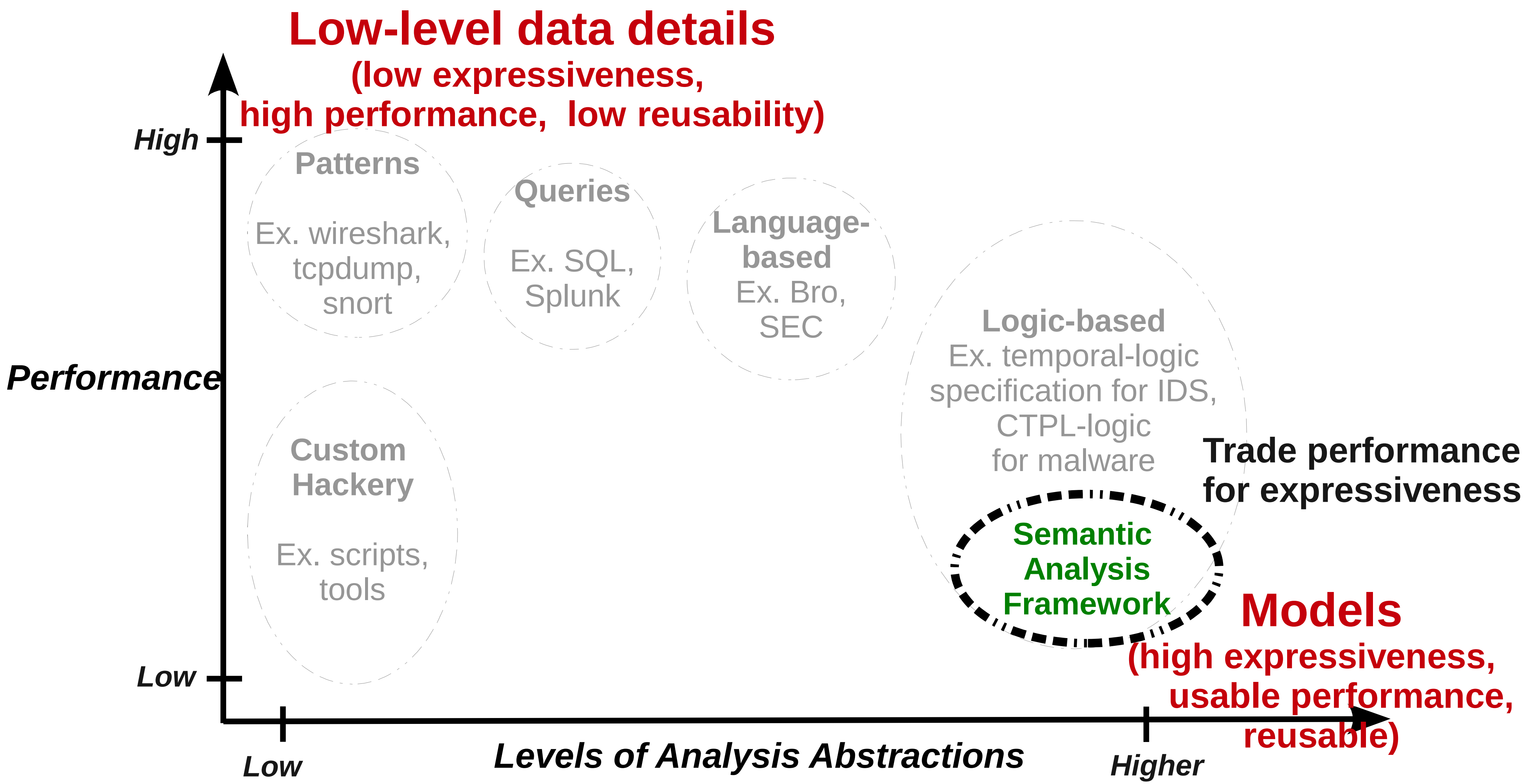


Models drive analysis over data!

Approximate Lay of the Land



Approximate Lay of the Land



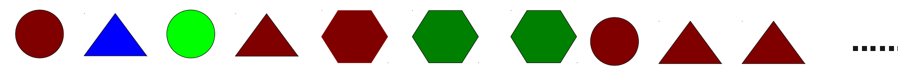
Key differences with other logic-based approaches

- **<u>Composable abstractions</u> to capture semantics**
- **<u>Expressive relationships</u> for networked systems**

Basics of our Modeling Approach

DATA
Multitype, multi-variate, timestamped

FACTS

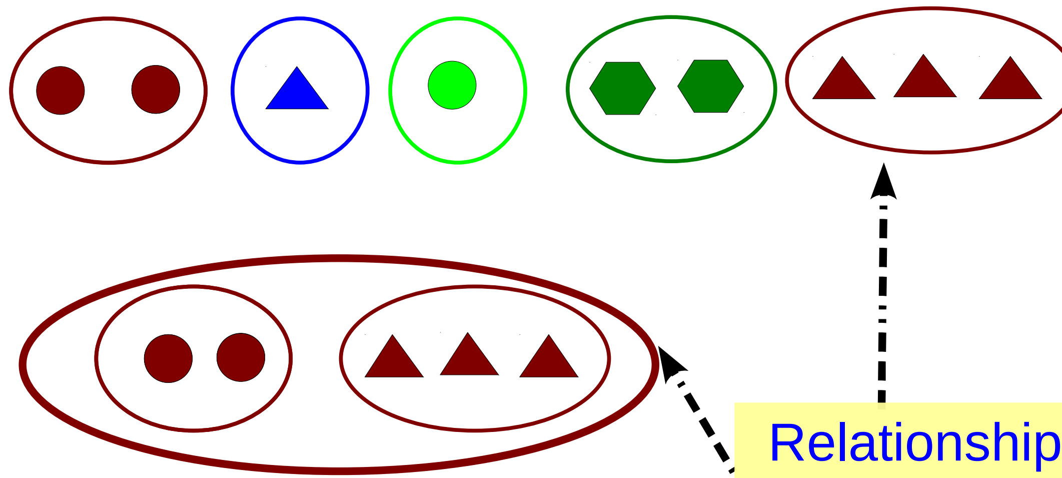


(ex: FILE_OPEN, FILE_CLOSE, TCP_PACKET, ...

Behavior
(fundamental abstraction)
Sequence or group of one or more **related** facts

Complex Behaviors
Related behaviors

Model
Top-level behavior



Relationships are key

Models encode higher-level system semantics!

Relationships in the Modeling Language

Temporal Relationships

Causality/Ordering
Eventuality
Invariance
Synchrony/Timing

A file open **eventually leads to** a file close

Temporal Operators

FILE_OPEN **~>** *FILE_CLOSE*

Concurrent Relationships

Parallelism
Overlaps

HTTP and FTP flows are **concurrent**.

Interval Temporal Operators

HTTP_FLOW ***olap*** *FTP_FLOW*

Logical Relationships

Combinations
Exclusions

Experiment **either** succeeds **or** fails

Logical Operators

EXPT_SUCCESS ***xor*** *EXPT_FAIL*

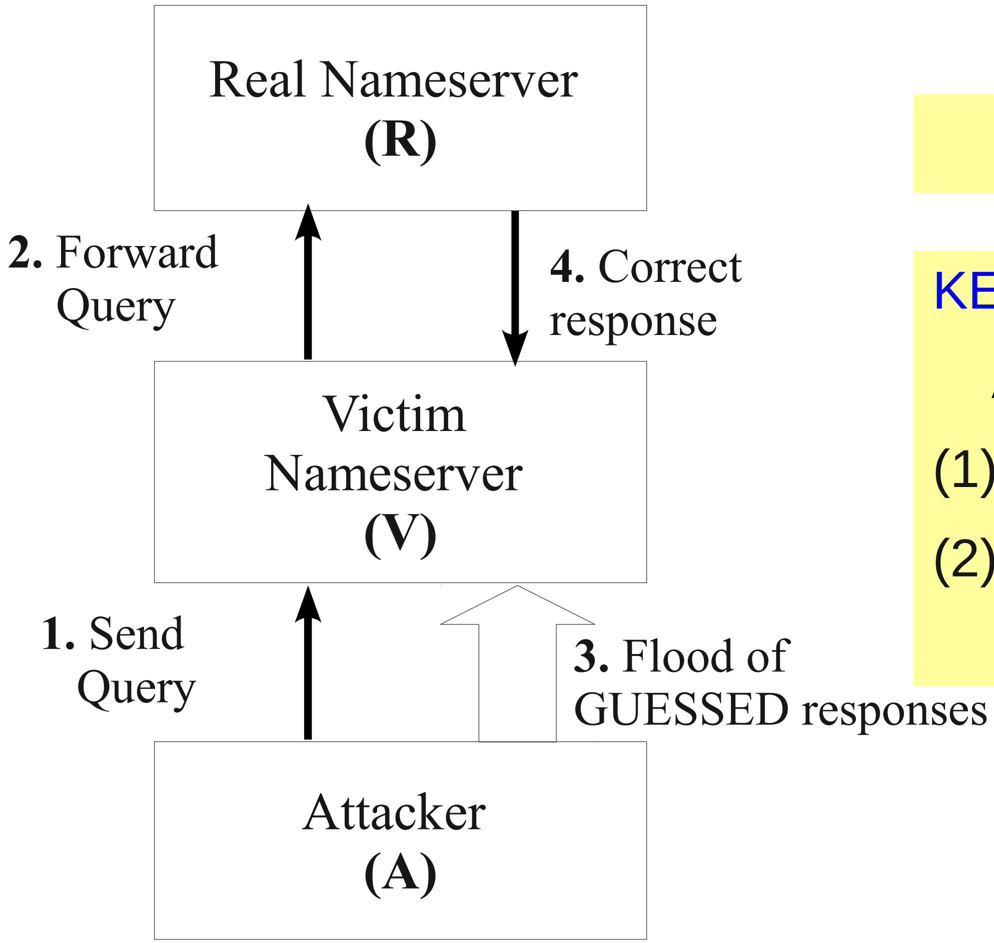
Dependency relationships b/w data attributes

File open and file close are behaviors **related by their filename**.

FILE_CLOSE.name = FILE_OPEN.name

Cache Poisoning Behavior

Cache Poisoning Behavior (DNS Kaminsky)



Objective: Attacker poisons the victim's DNS cache.

Steps 1-4 keep running in a loop.

KEY ISSUES

Attacker fails to poison cache due to

(1) Race conditions with real nameserver.

(2) Incorrectly GUESSED responses.

Analysis using typical approach

```
08:06:30.151588 IP 10.1.2.2.domain > 10.1.4.2.32771: 8745 NXDomain* 0/1/1 (118)
08:06:30.151602 IP 10.1.2.2.domain > 10.1.4.2.32771: 37972* 1/0/1 NS B.ROOT-SERVERS.NET. (59)
08:06:30.151823 IP 10.1.4.2.domain > 10.1.11.2.32772: 45918 NXDomain 0/1/0 (107)
08:08:45.324938 IP 10.1.11.2.32772 > 10.1.4.2.domain: 20572+ PTR? 1.253.168.192.in-addr.arpa. (44)
08:08:45.325058 IP 10.1.4.2.domain > 10.1.11.2.32772: 20572 NXDomain 0/1/0 (107)
08:09:09.787858 IP 10.1.11.2.32772 > 10.1.4.2.domain: 0 NS? bofa.com. (26)
```

Tricky to analyze

- Requires Expertise.
- Too many random values in the data to extract using simple patterns.
- Race conditions (timing issues) are hard to debug over 10's of thousands of packets.
- Many ways to fail.

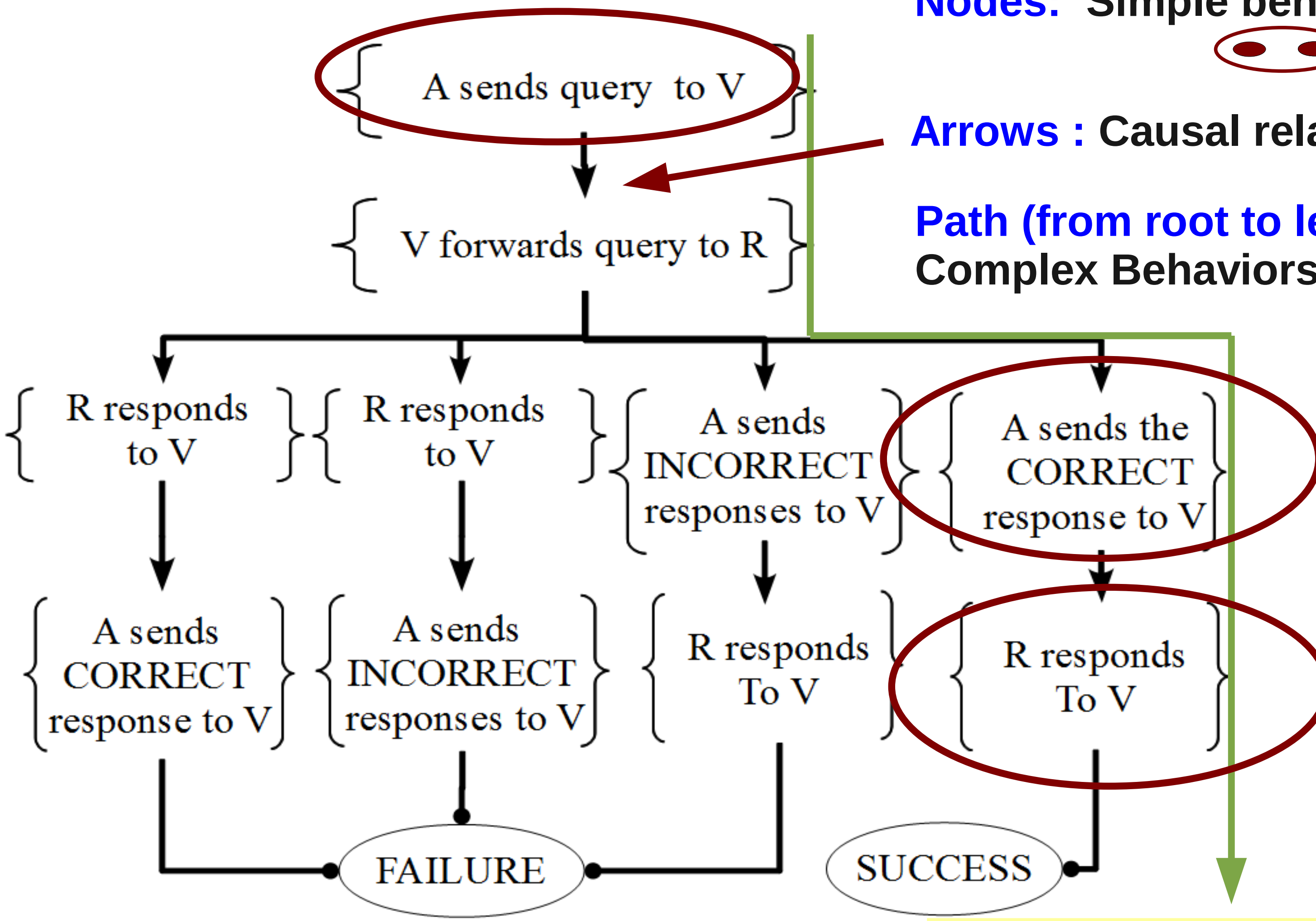
```
08:09:11.185802 IP 10.1.6.3.domain > 10.1.4.2.32778: 2228*- 1/1/1 A 159.16.126.233 (103)
08:09:11.186301 IP 10.1.6.3.domain > 10.1.4.2.32778: 2228*- 1/1/1 A 159.16.126.233 (103)
08:09:11.186551 IP 10.1.6.3.domain > 10.1.4.2.32778: 2228*- 1/1/1 A 159.16.126.233 (103)
08:09:11.186812 IP 10.1.6.3.domain > 10.1.4.2.32778: 2228*- 1/1/1 A 159.16.126.233 (103)
08:09:11.187051 IP 10.1.6.3.domain > 10.1.4.2.32778: 2228*- 1/1/1 A 159.16.126.233 (103)
```

Model of Behavior

Nodes: Simple behavior

Arrows : Causal relationships

Path (from root to leaf) :
Complex Behaviors



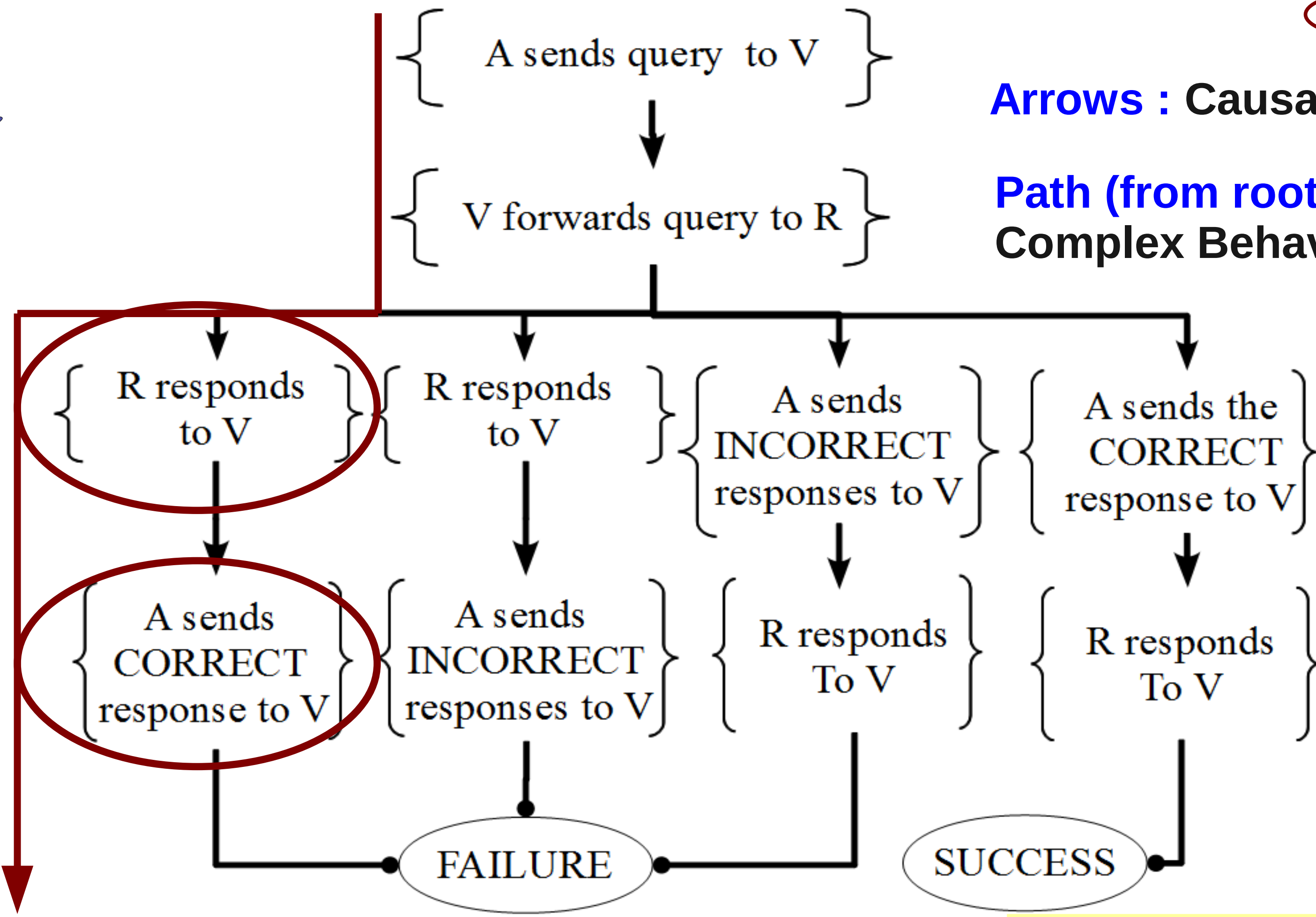
SUCCESS =
A guesses right and wins race with R

Model of Behavior

Nodes: Simple behavior

Arrows : Causal relationships

Path (from root to leaf) : Complex Behaviors



TIMING_FAIL =
A guesses right but
loses race to R.

SUCCESS =
A guesses right and
wins race with R

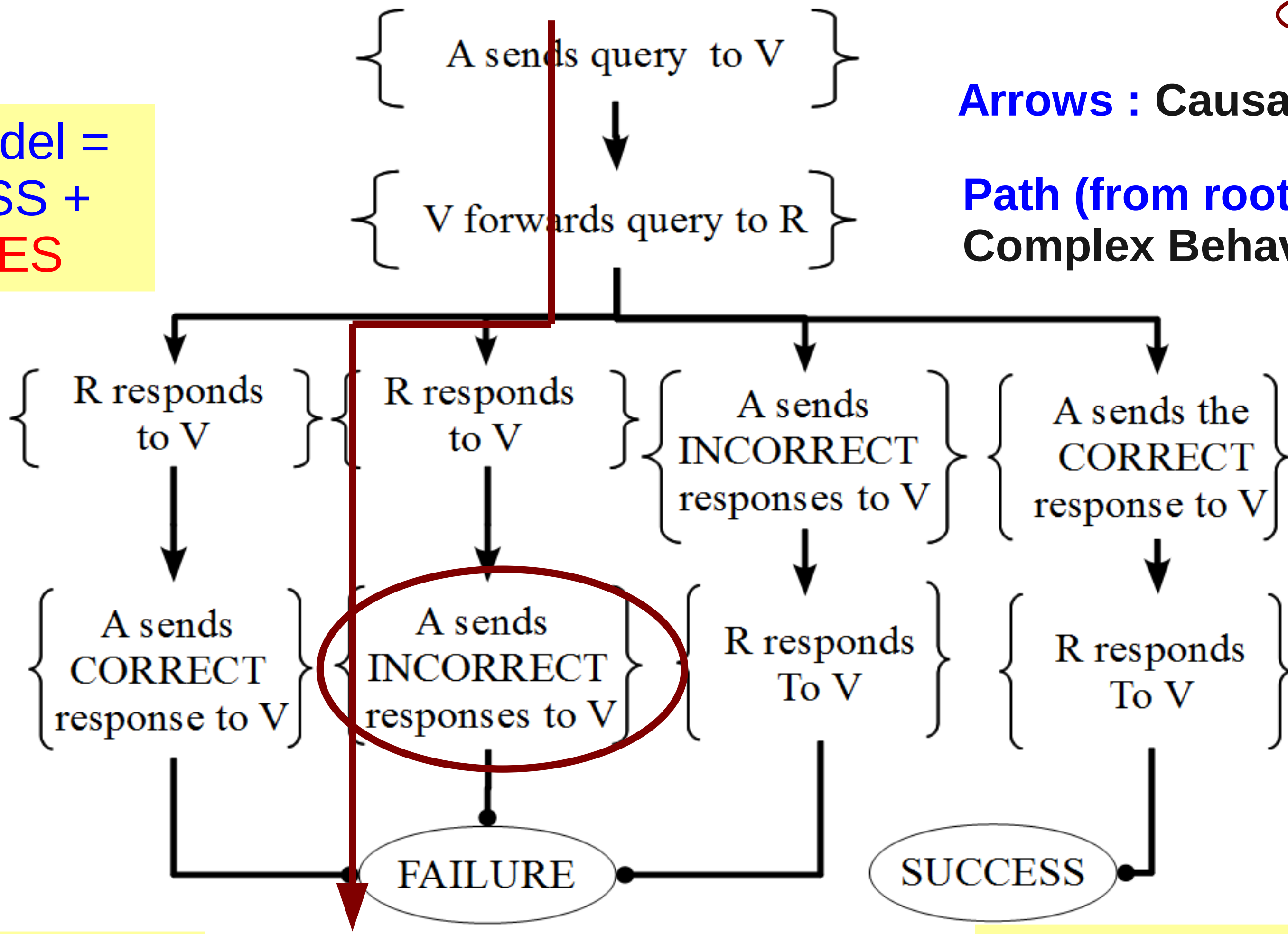
Model of Behavior

Behavior Model =
1 SUCCESS +
3 FAILURES

Node: **Simple behavior**

Arrows : **Causal relationships**

Path (from root to leaf) :
Complex Behaviors

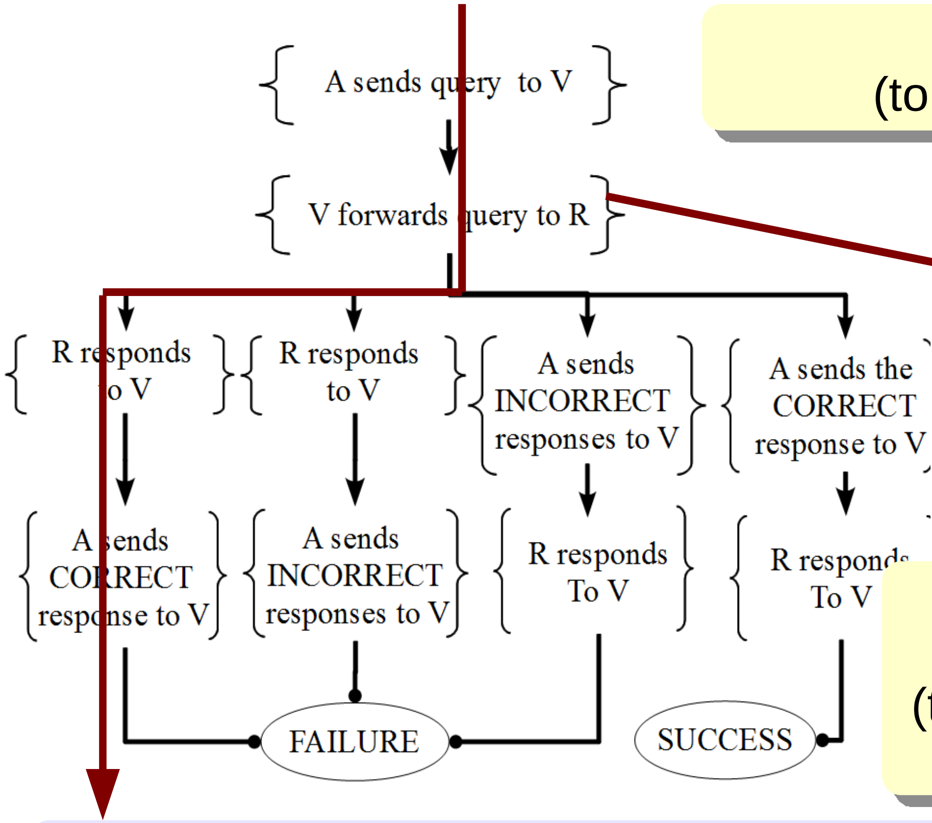


TIMING_FAIL =
A guesses right but
loses race to R.

BADGUESS_1 =
A guesses wrong
response

SUCCESS =
A guesses right and
wins race with R

Encoding the Model



#1. Capture simple behaviors
(to capture facts for each distinct attack step)

```
VtoR_query =
DNSREQRES.dns_req(sip=$AtoV_query.dip,
        dnsquesname=$AtoV_query.dnsquesname)
```

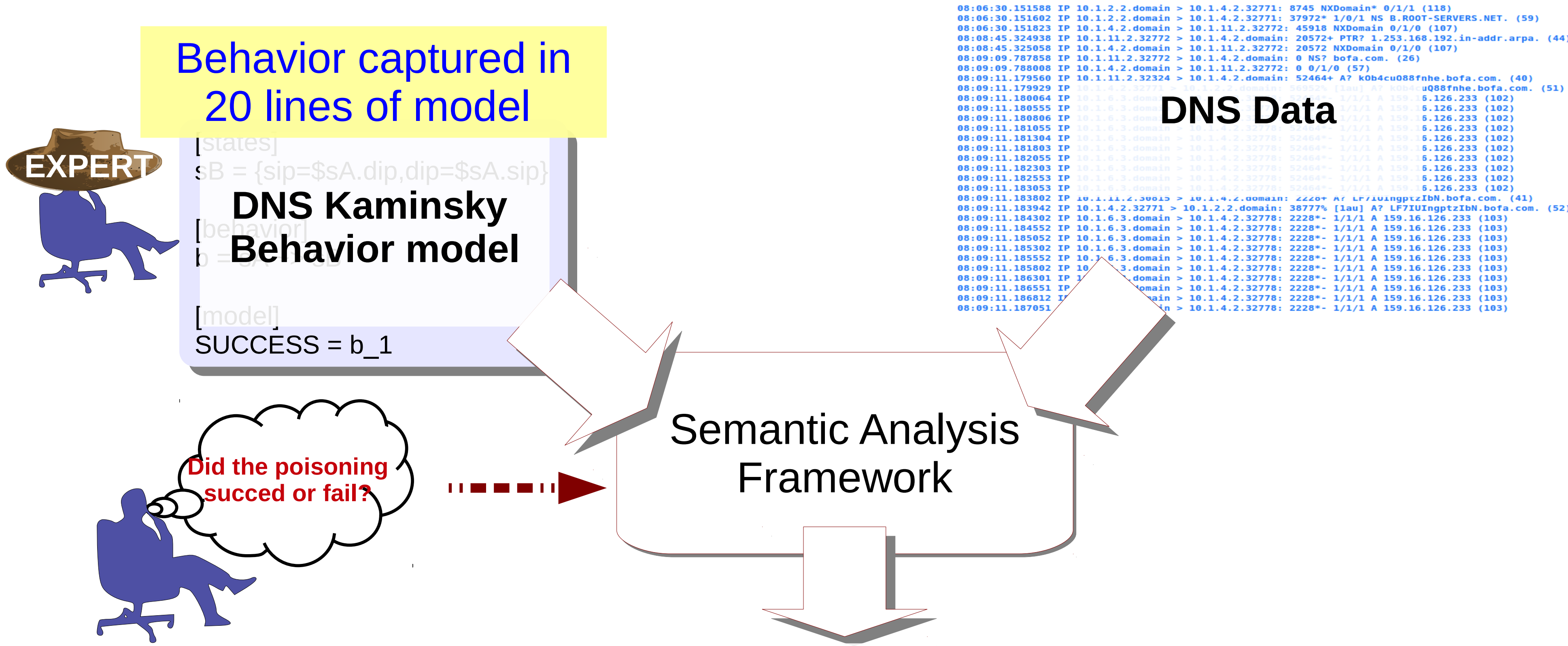
#2. Relate simple behaviors to form complex behaviors
(to capture the causal relationships between steps)
4 behaviors = 1 SUCCESS + 3 FAILURES

```
TIMING_FAIL = (AtoV_query ~> VtoR_query ~> RtoV_resp ~>AtoV_resp)
```

#3. Define Behavior Model
(assertion to capture users understanding of system operation)

```
DNSKAMINSKY = SUCCESS xor TIMING_FAIL xor BADGUESS_1 xor BADGUESS_2
```

Analysis Using the Model



Answers in the form of facts satisfying the model.

```
Summary : DNSCACHEPOISON_TIMING_FAIL
========================
Total Matching Instances: 622
  etype    | timestamp  | sip        | dip        | sport | dport | dnsid |   dnsauth
-------------------------------------------------------------------------------------------
  PKT_DNS | 1275515486 | 10.1.11.2 | 10.1.4.2 |  6916 |  53   | 47217 |
  PKT_DNS | 1275515486 | 10.1.4.2  | 10.1.6.3 | 32778 |  53   | 15578 |
  PKT_DNS | 1275515486 | 10.1.6.3  | 10.1.4.2 |  53   | 32778 | 15578 |realns.eby.com
  PKT_DNS | 1275515486 | 10.1.6.3  | 10.1.4.2 |  53   | 32778 | 47217 |fakens.fakeeby.com
  PKT_DNS | 1275515486 | 10.1.6.3  | 10.1.4.2 |  53   | 32778 | 47217 |fakens.fakeeby.com
  PKT_DNS | 1275515486 | 10.1.6.3  | 10.1.4.2 |  53   | 32778 | 47217 |fakens.fakeeby.com
  PKT_DNS | 1275515486 | 10.1.6.3  | 10.1.4.2 |  53   | 32778 | 47217 |fakens.fakeeby.com
  PKT_DNS | 1275515486 | 10.1.6.3  | 10.1.4.2 |  53   | 32778 | 47217 |fakens.fakeeby.com
  PKT_DNS | 1275515486 | 10.1.6.3  | 10.1.4.2 |  53   | 32778 | 47217 |fakens.fakeeby.com
  PKT_DNS | 1275515486 | 10.1.6.3  | 10.1.4.2 |  53   | 32778 | 47217 |fakens.fakeeby.com
```

TIMING_FAIL
(A loses the race against R)

Current Implementation and Performance

- **Prototype algorithm for applying models over data.**
- **Algorithm performance**
  - $O(N^2)$ worst-case performance
  - Straight-forward
- **Analysis Framework**
  - Written in Python
  - SQLite-based storage backend
- Scalability and performance issues are under active investigation.

Applicability

- Broad range of event-based modeling in networked systems
- More examples in paper
  - Modeling hypotheses
    - Ex. Validating DoS detection heuristics over traces
  - Modeling a security threat
    - Ex. Model of a simple worm spread over IDS logs
  - Modeling dynamic change
    - Ex. Model of changes in traffic rate due to attack.

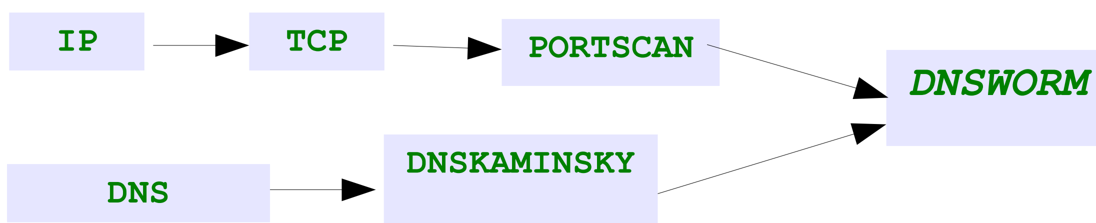
Future Work

- Extend Modeling Capabities
  - Modeling probabilistic behavior
  - Modeling packet distributions
- Analysis Framework
  - Scalability and performance
  - Reducing the computational complexity of correlations using dependent attributes.

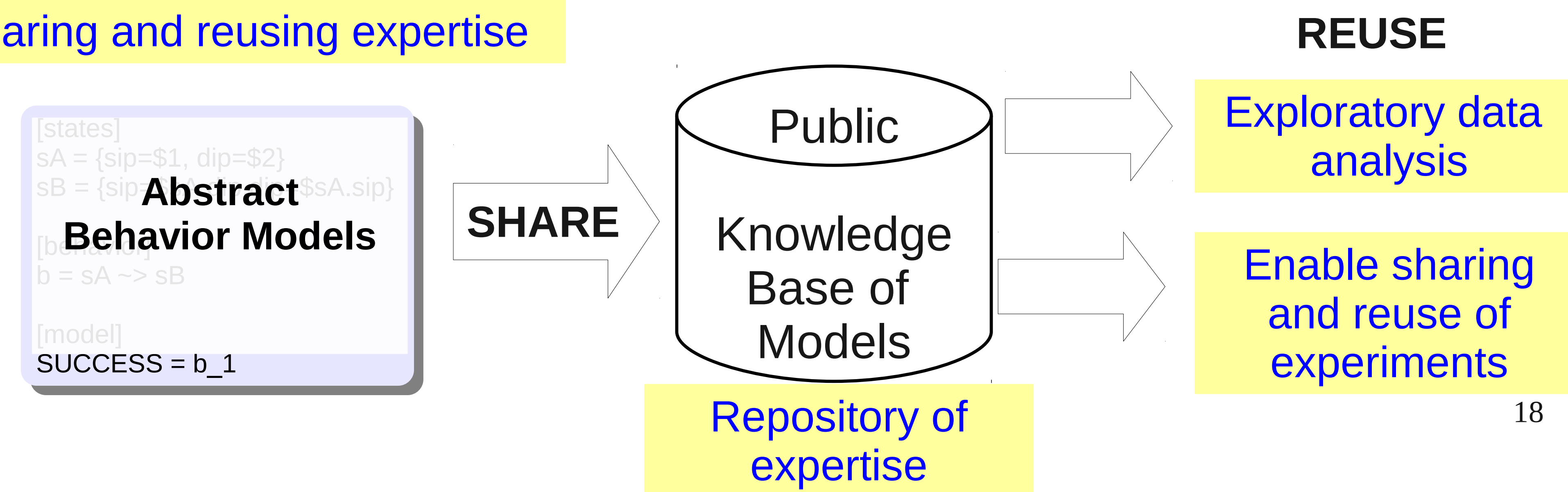
Composing, Sharing and Reusing

Semantic Analysis Framework enables data analysis at higher-levels of abstraction.

Composing models to create higher-level meaning



Sharing and reusing expertise



Thank You!

Our framework will soon be publicly available at
http://thirdeye.isi.deterlab.net

Please register on our mailing-list to stay in tune with release and updates