ETH zürich
Department of Computer Science

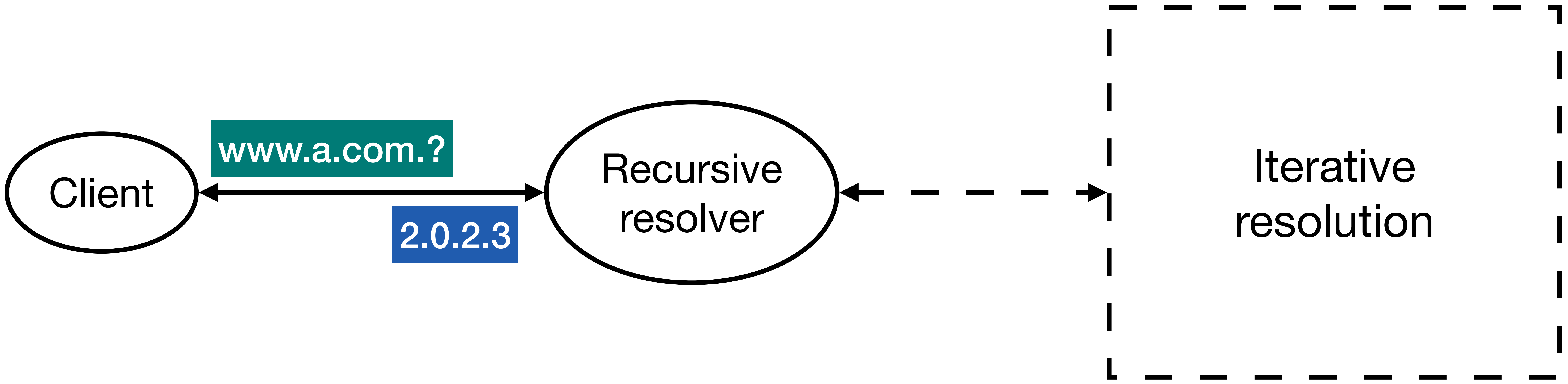
A Formal Framework for End-to-End DNS Resolution

Si Liu*, Huayi Duan*, Lukas Heimes, Marco Bearzi, Jodok Vieli,
David Basin, and Adrian Perrig
*(co-first author)

SIGCOMM 2023, New York

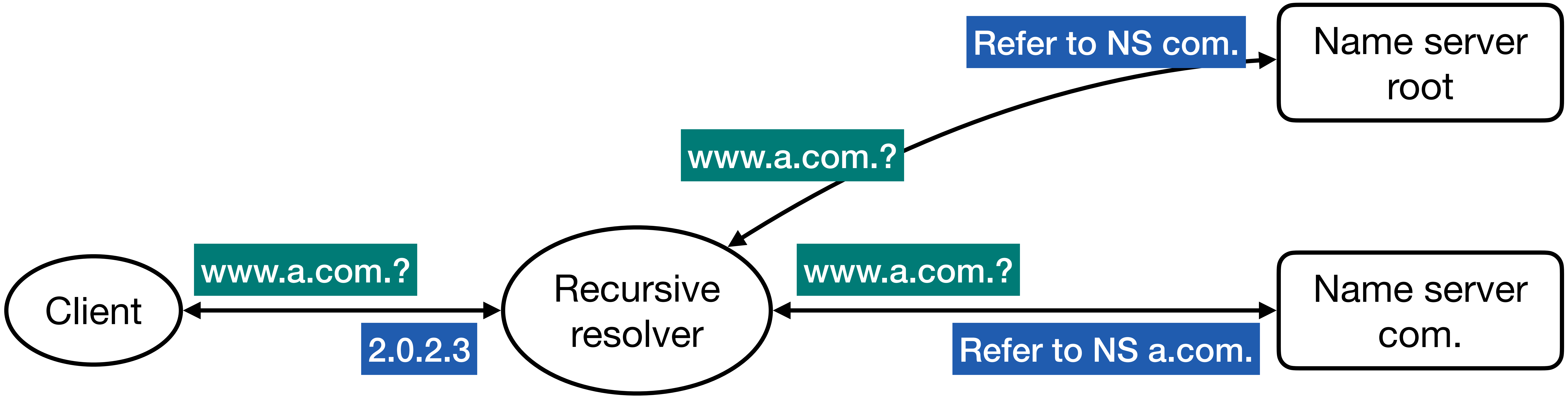
DNS is more complex than you might think

Textbook example for name resolution



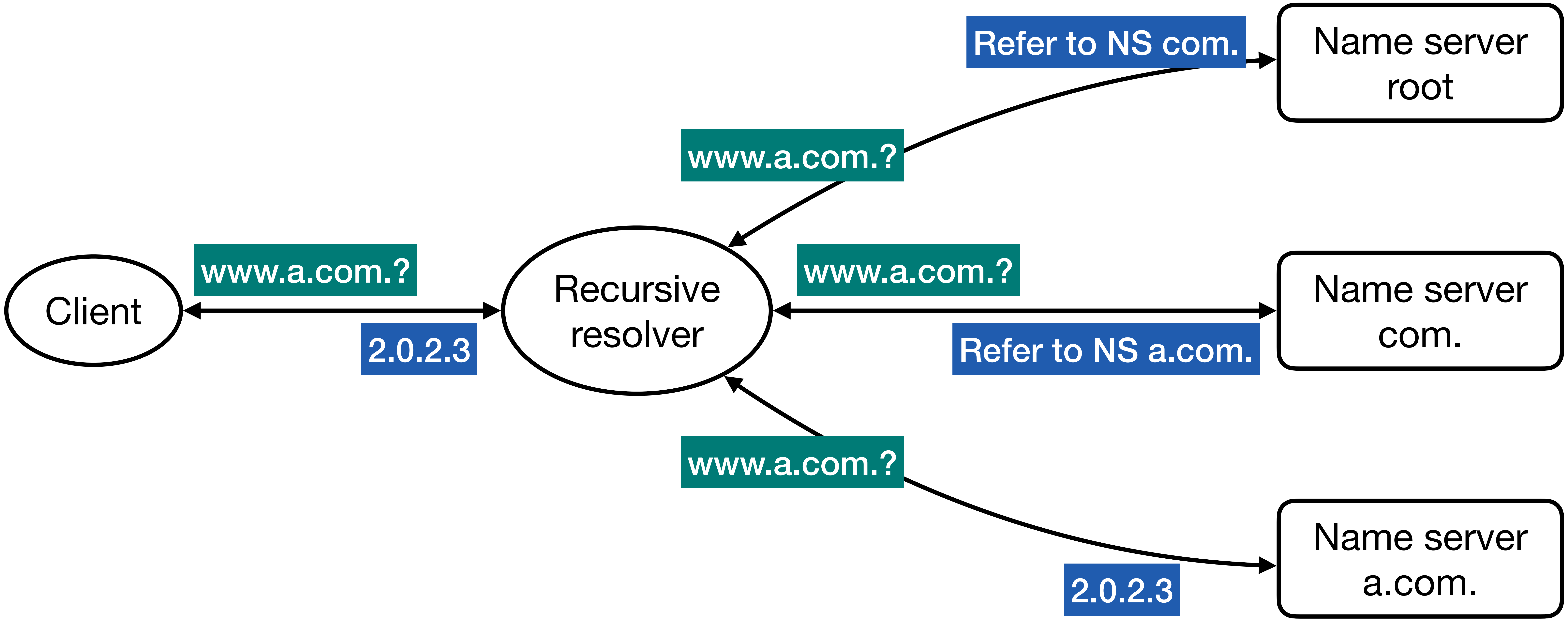
DNS is more complex than you might think

Textbook example for name resolution



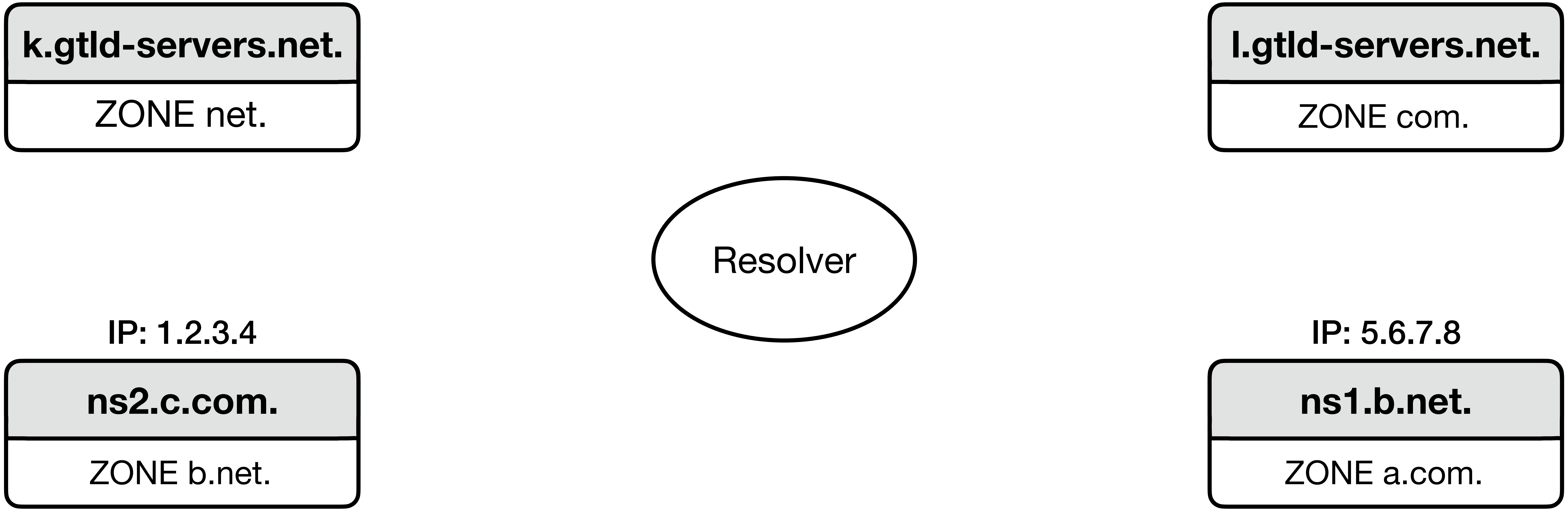
DNS is more complex than you might think

Textbook example for name resolution



DNS is more complex than you might think

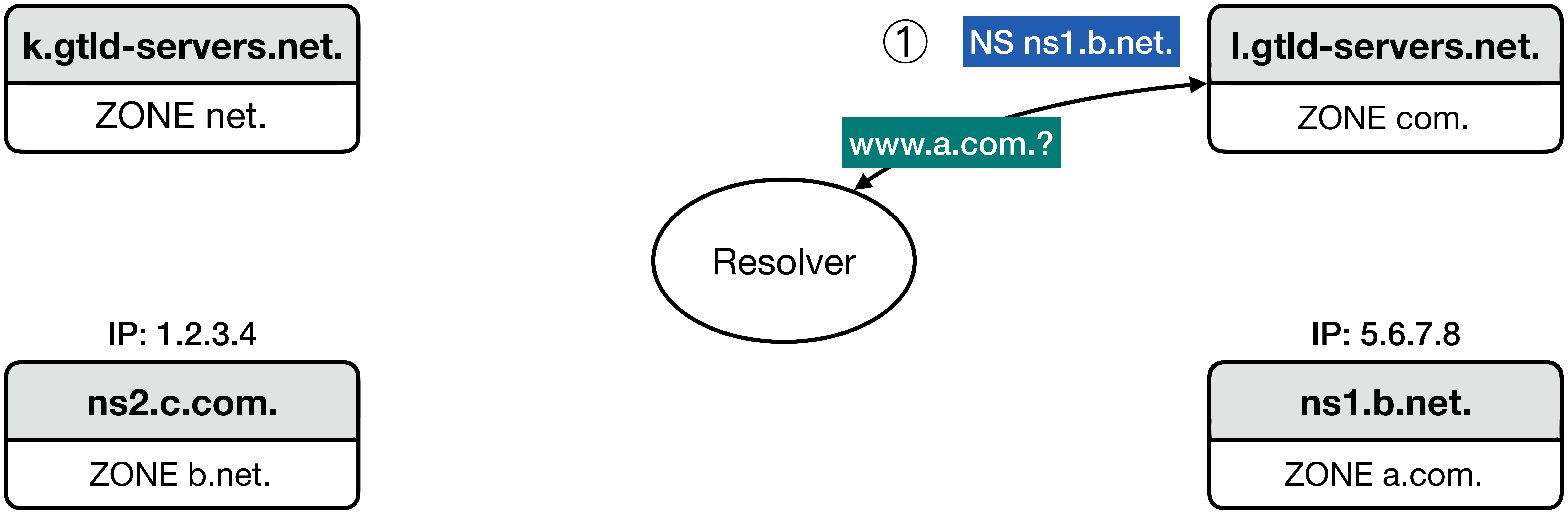
Reality: subqueries to resolve referrals



*Ignore root NS, assume TLD NS addresses are known

DNS is more complex than you might think

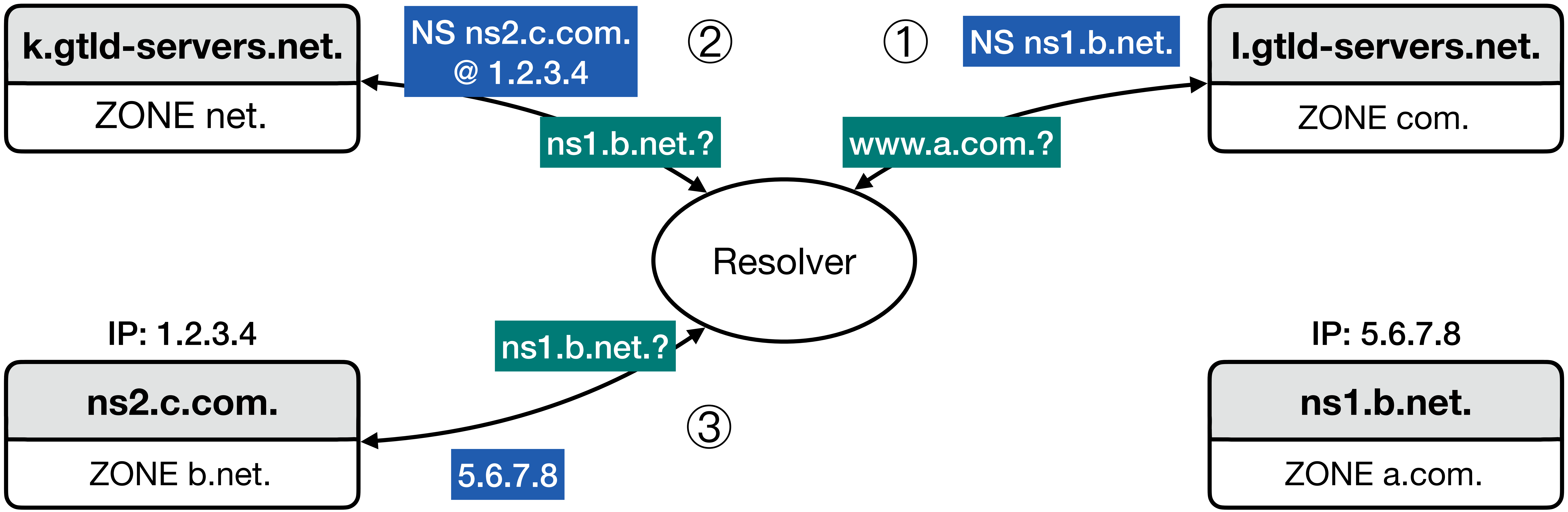
Reality: subqueries to resolve referrals



*Ignore root NS, assume TLD NS addresses are known

DNS is more complex than you might think

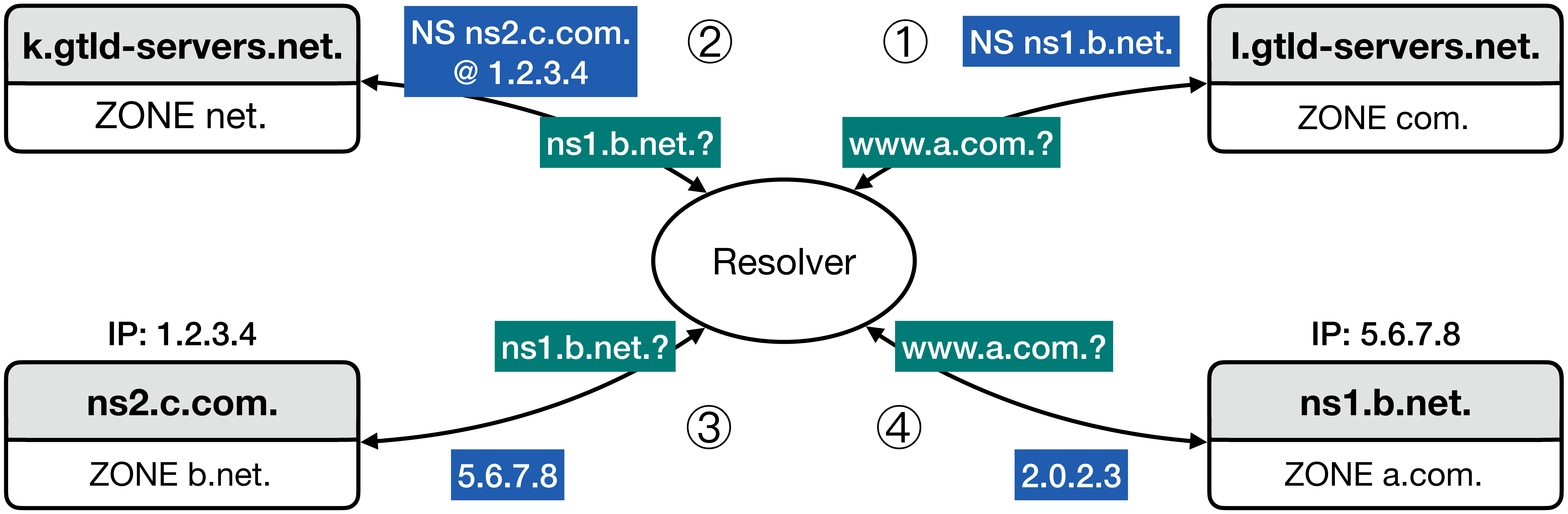
Reality: subqueries to resolve referrals



*Ignore root NS, assume TLD NS addresses are known

DNS is more complex than you might think

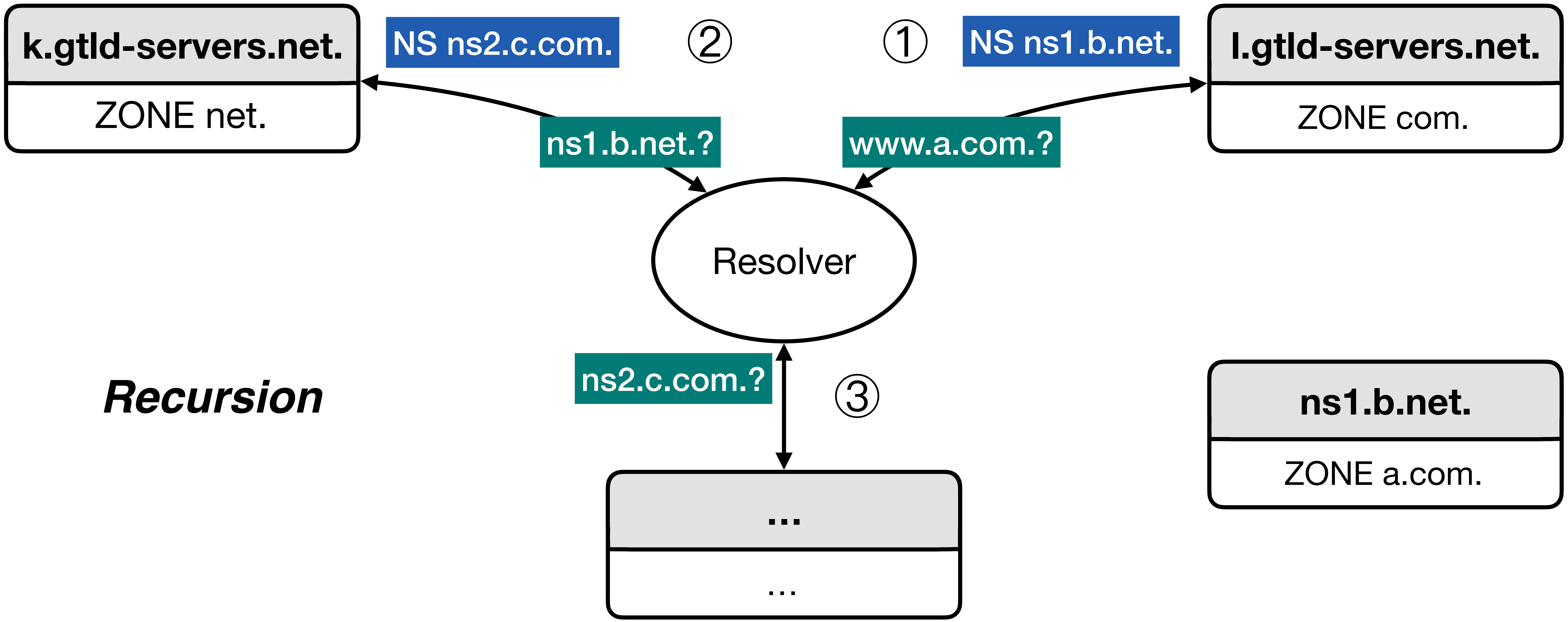
Reality: subqueries to resolve referrals



*Ignore root NS, assume TLD NS addresses are known

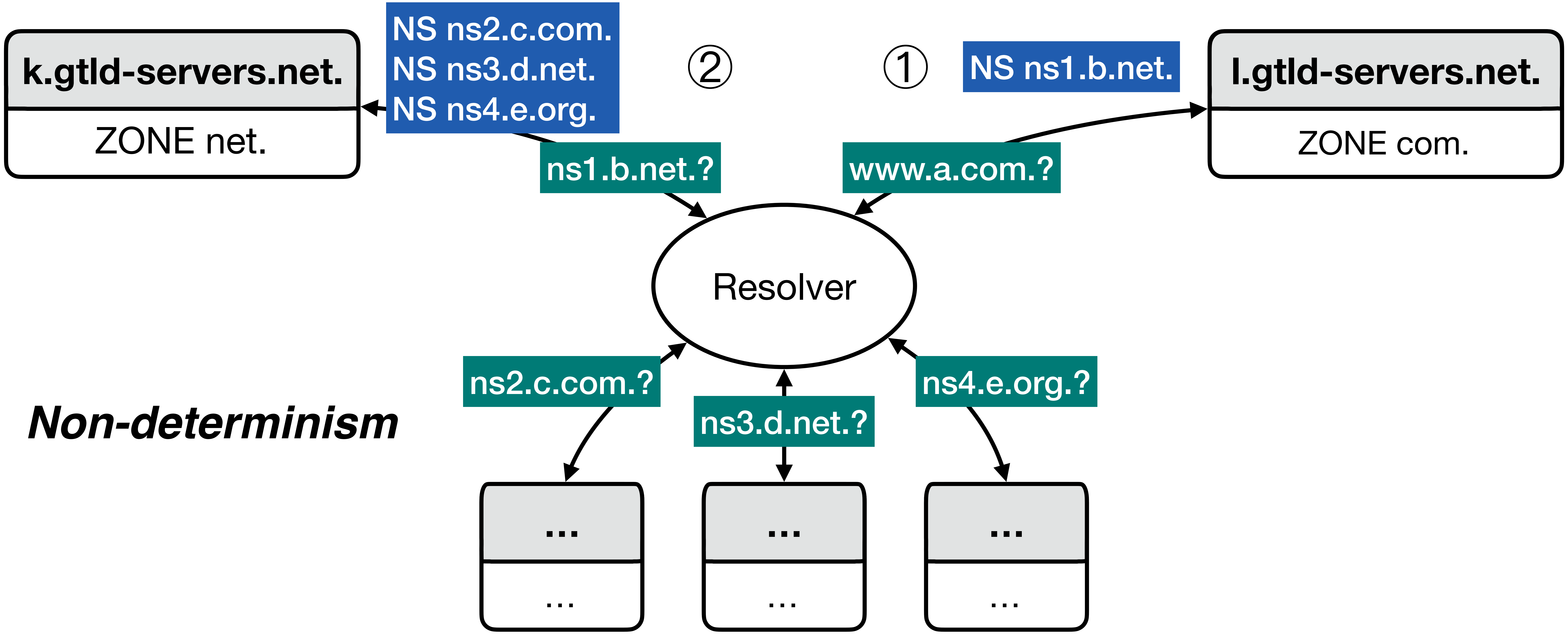
DNS is more complex than you might think

Reality: subqueries to resolve referrals



DNS is more complex than you might think

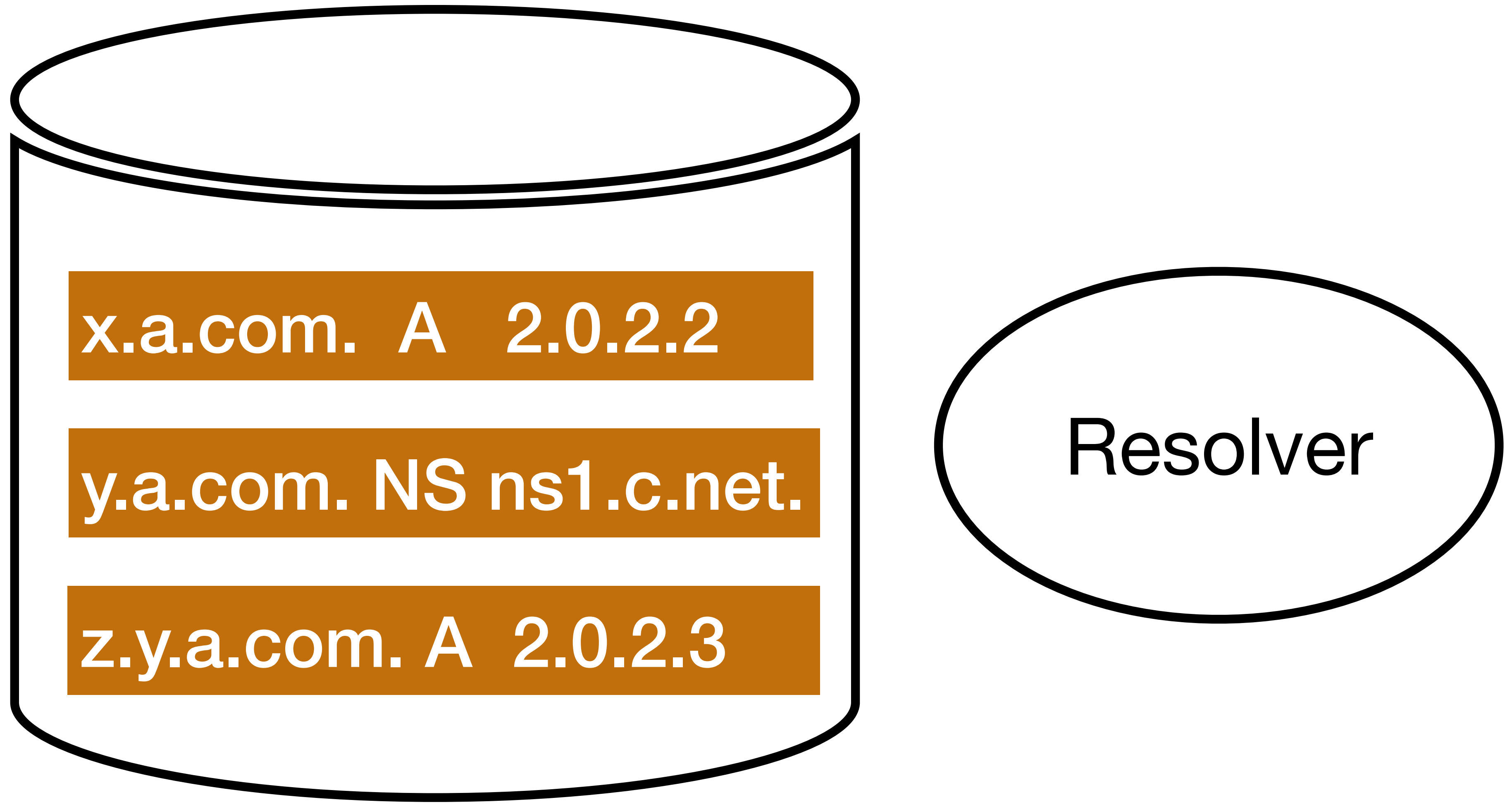
Reality: subqueries to resolve referrals



Non-determinism

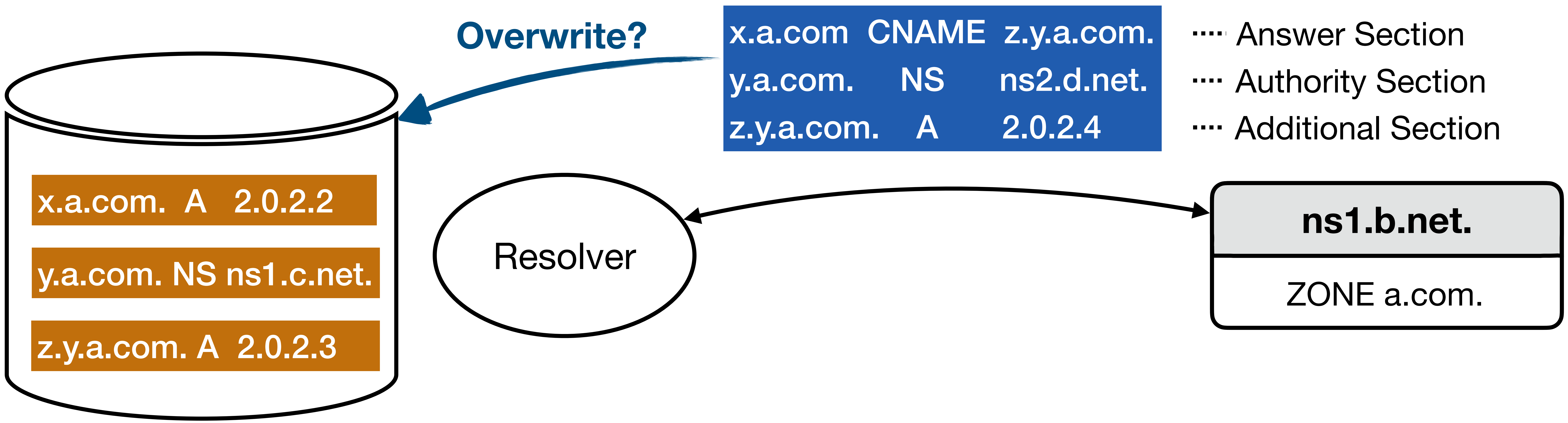
DNS is more complex than you might think

Reality: caching (TTL, positive/negative, concurrency, data credibility, …)



DNS is more complex than you might think

Reality: caching (TTL, positive/negative, concurrency, data credibility, …)



DNS is more complex than you might think

Reality: subqueries, query rewrite, caching, …

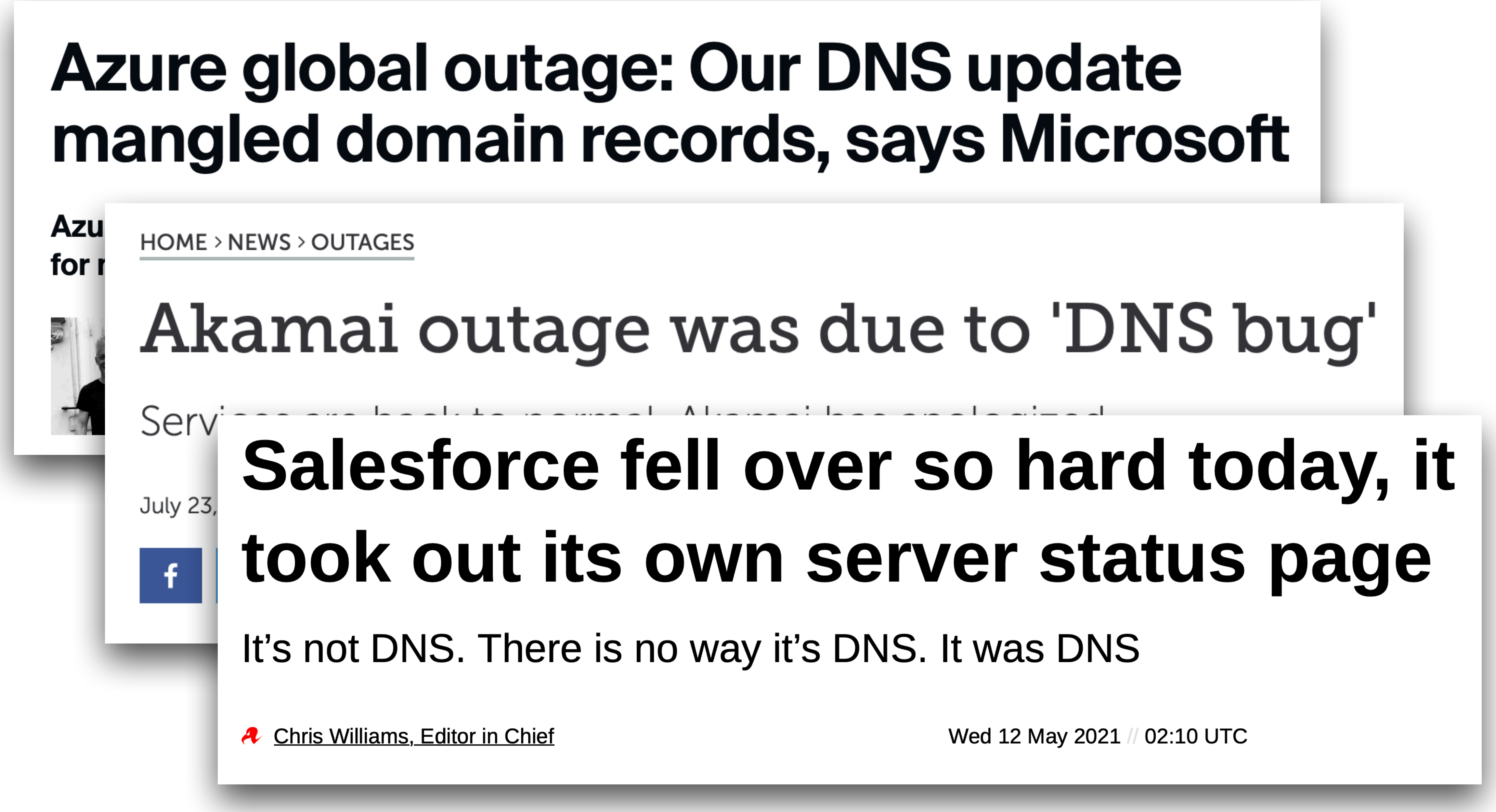
“DNS Camel” and beyond:

- Over ***300*** RFCs
- Over ***5000*** pages
- Growing at ***~2 pages / week***

DNS features are ***entangled***!

Unsurprisingly, DNS bugs and vulnerabilities prevail

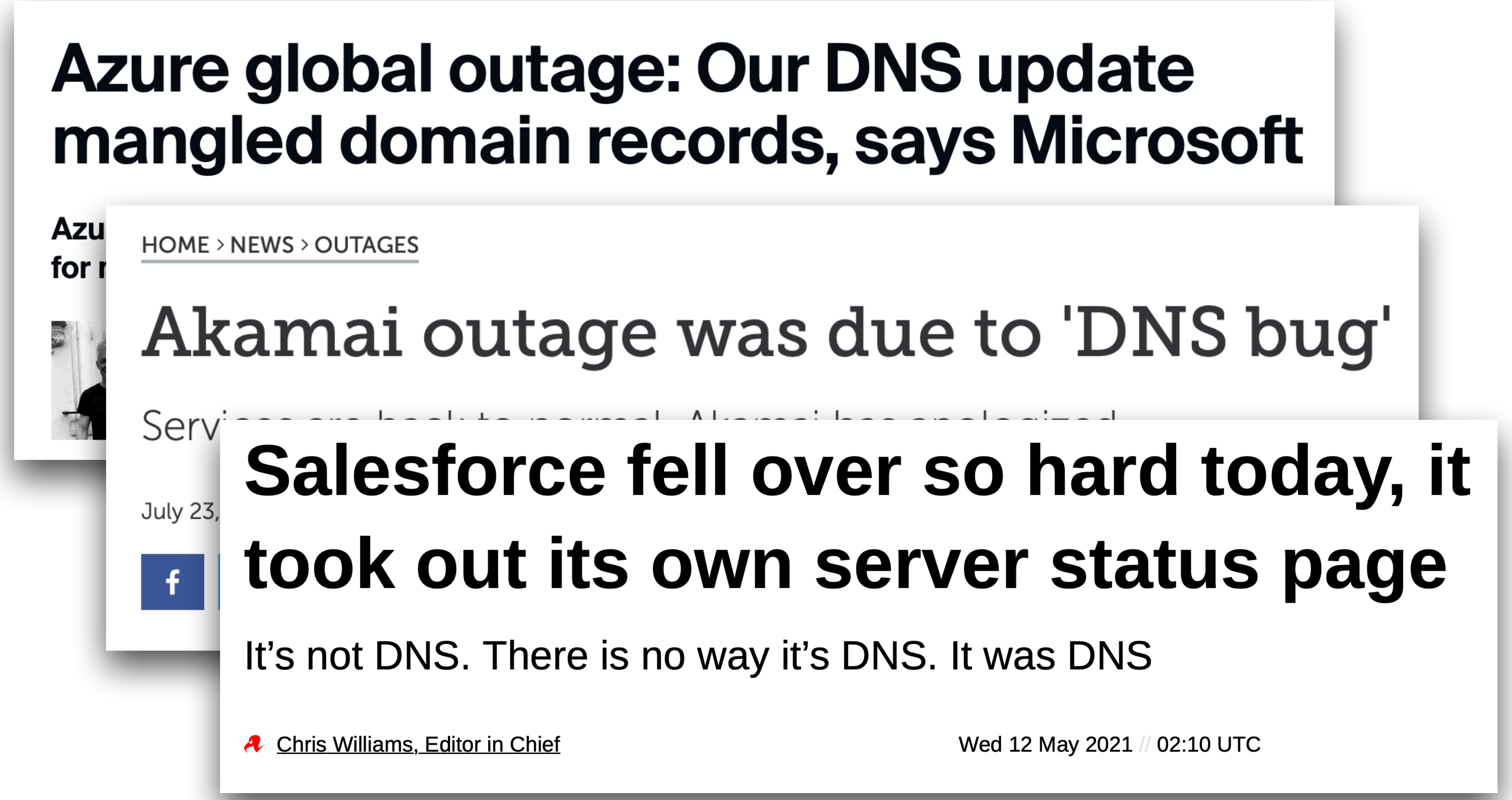
Frequent outages due
to **misconfigurations**



Sources:
- https://www.zdnet.com/
- https://www.datacenterdynamics.com/
- https://www.theregister.com/

Unsurprisingly, DNS bugs and vulnerabilities prevail

Frequent outages due to **misconfigurations**


Azure global outage: Our DNS update mangled domain records, says Microsoft

HOME › NEWS › OUTAGES

Akamai outage was due to 'DNS bug'

July 23,

Salesforce fell over so hard today, it took out its own server status page

It's not DNS. There is no way it's DNS. It was DNS

Chris Williams, Editor in Chief Wed 12 May 2021 // 02:10 UTC

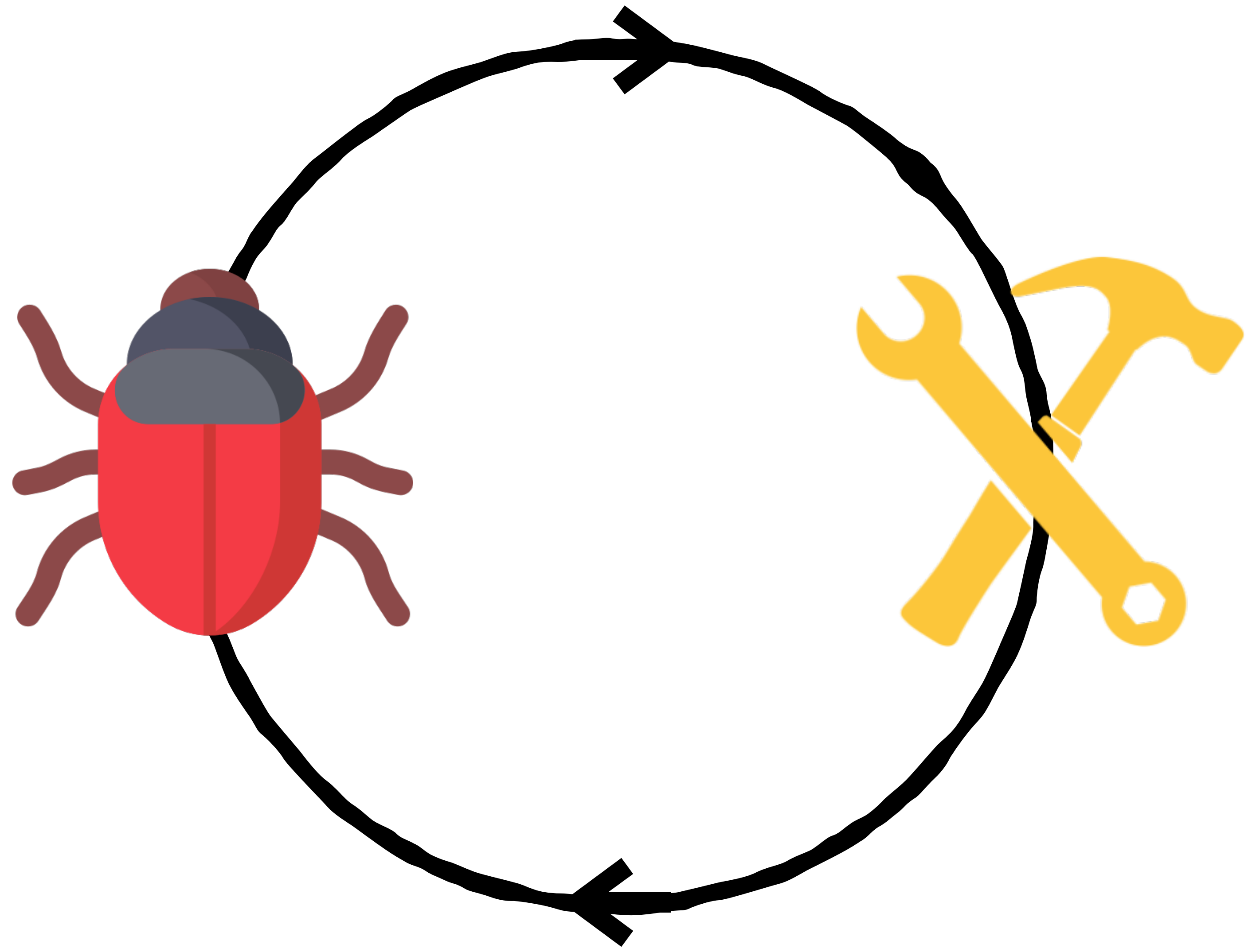
Sources:
- https://www.zdnet.com/
- https://www.datacenterdynamics.com/
- https://www.theregister.com/

Frequent discovery of **security vulnerabilities**

- Infinite delegation [DNS-OARC'15]
- Unchained [RAID'15]
- NXNS [SEC'20]
- Zaw [CCS'20]
- SADDNS [CCS'20, CCS'21]
- TsuNAME [IMC'21]
- MaginotDNS [SEC'23]
- NRDelegation [SEC'23]
- PHOENIX DOMAIN [NDSS'23]
- …

Reasoning about DNS requires a principled approach

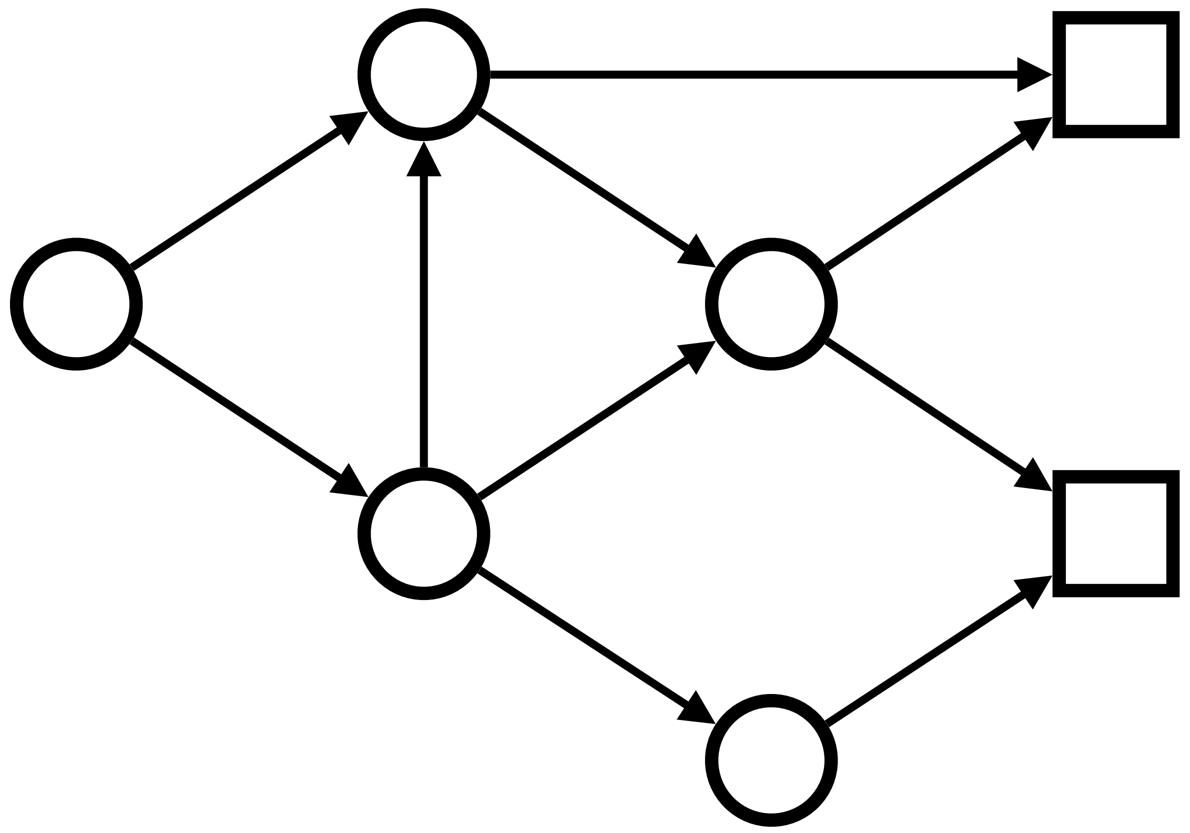
Break-and-Fix is insufficient

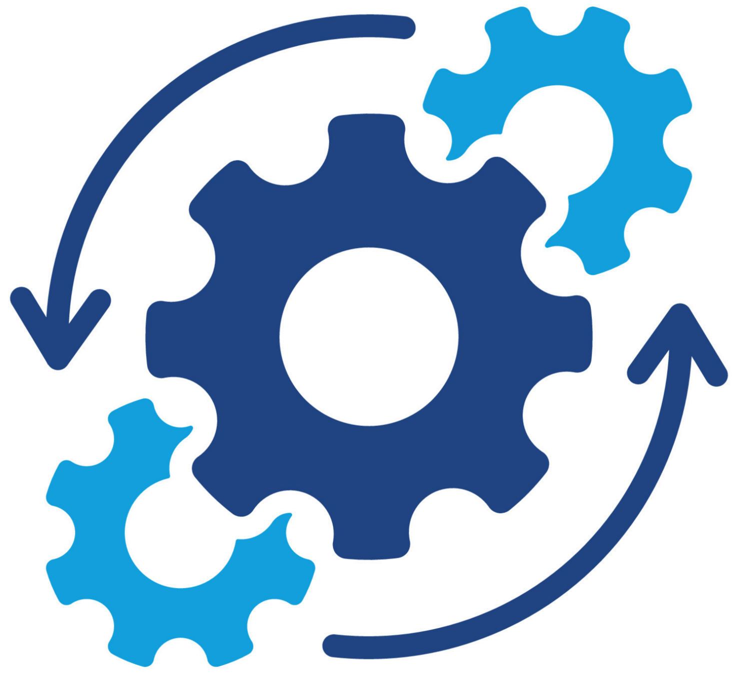


Reasoning about DNS requires a principled approach

Break-and-Fix is insufficient

Need **proactive**, **systematic** & **automated** analyses





Reasoning about DNS requires a principled approach

Break-and-Fix is insufficient

Need **proactive**, **systematic** & **automated** analyses

on a **mathematically precise** DNS model

RFCs are written in ***natural language*** with **ambiguities** and **underspecifications** … lead to problems!

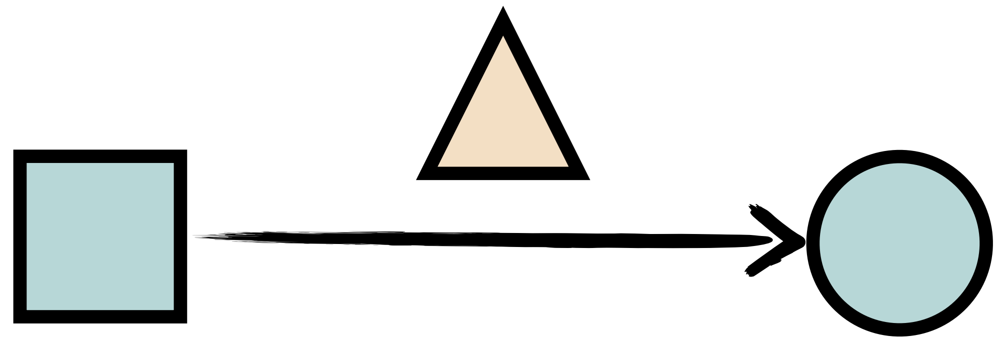
Our framework — modeling language and scope

Maude: a formal language supporting

✓ Expressive formalism based on ***rewriting logic***

✓ ***Concurrent*** computation with ***state***

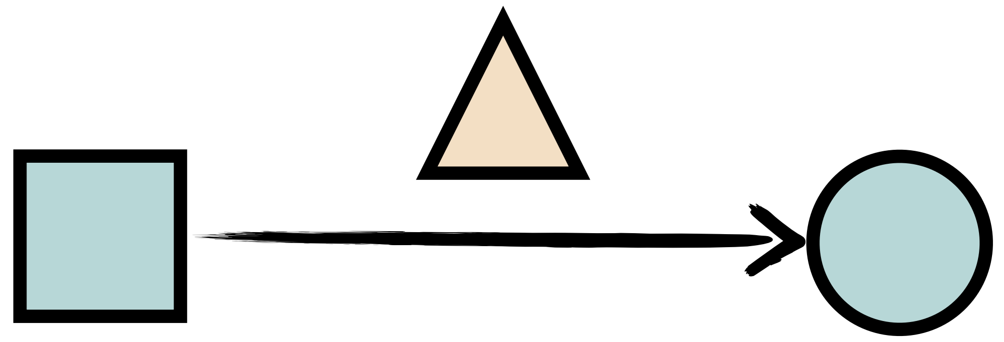
✓ ***Extensive*** tools for formal specification & verification



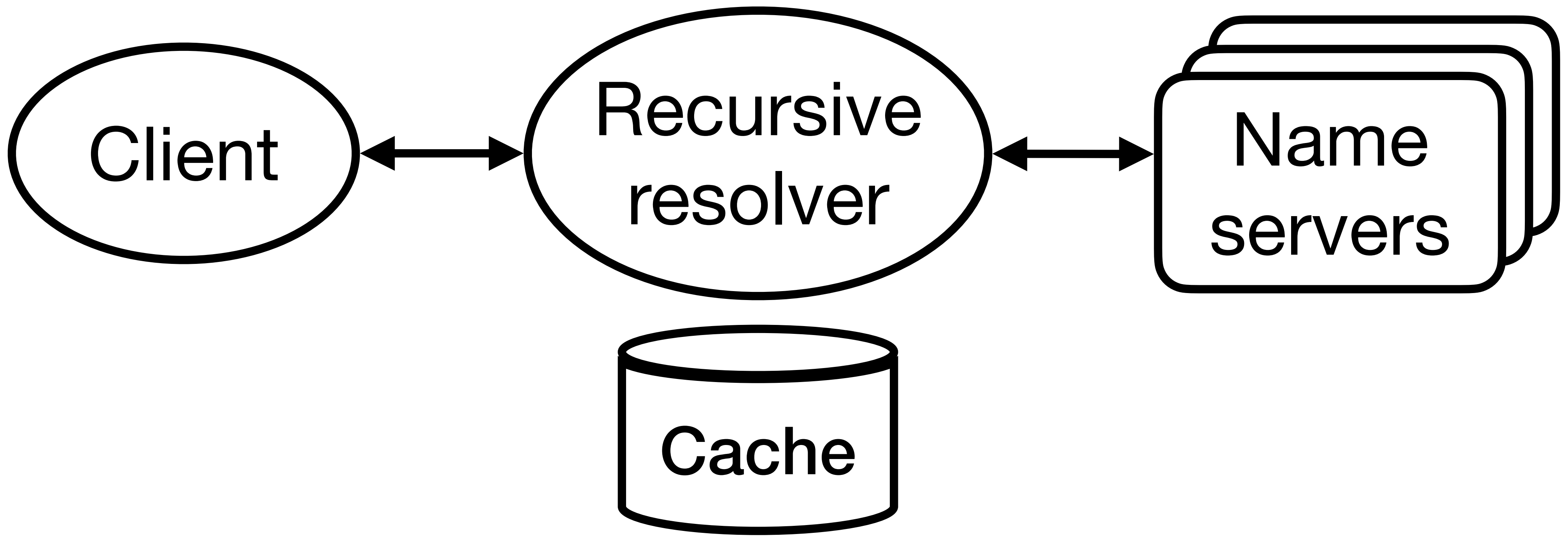
Our framework — modeling language and scope

Maude: a formal language supporting

- ✓ Expressive formalism based on ***rewriting logic***
- ✓ ***Concurrent*** computation with ***state***
- ✓ ***Extensive*** tools for formal specification & verification



Our scope: ***end-to-end*** name resolution up to the ***latest*** algorithmic refinements in RFC9156



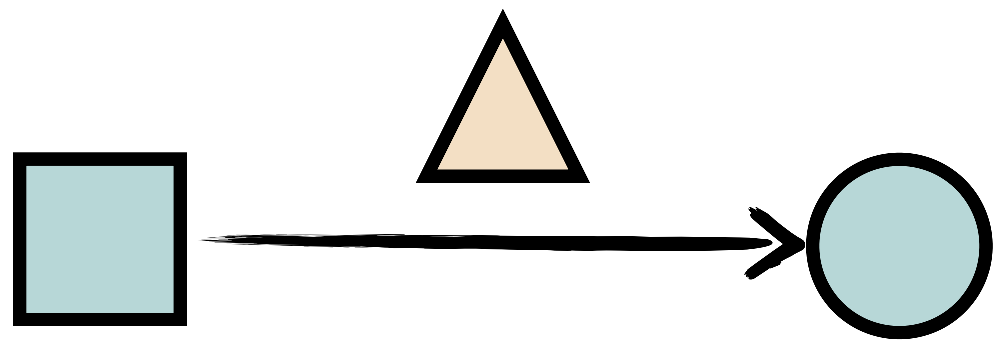
Our model's RFC coverage

RFC	Description
1034 [36]	Core specification
1035 [37]	Core specification
2181 [21]	Clarifications
2308 [6]	Negative caching
4592 [29]	Wildcards
6604 [25]	RCODE clarifications
6672 [40]	DNAME redirection
8020 [12]	NXDOMAIN clarification
9156 [11]	QNAME minimization

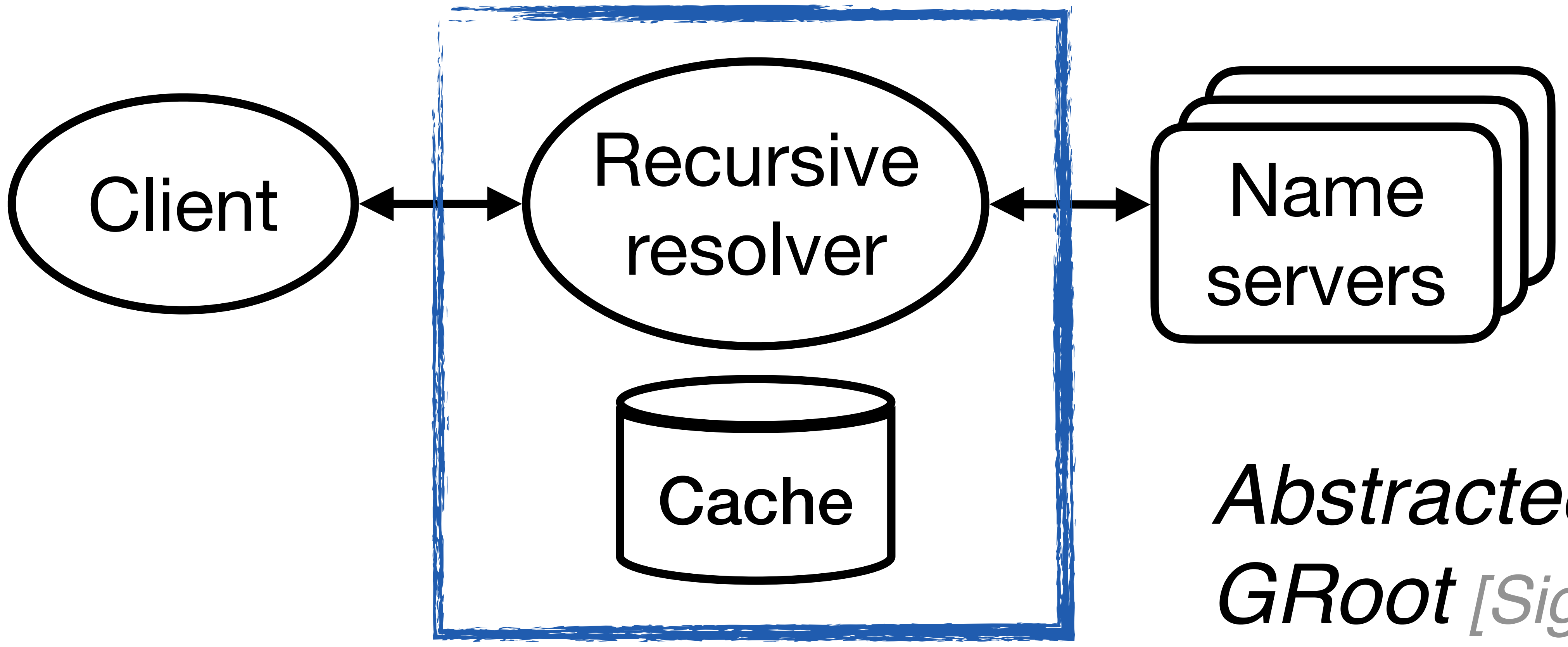
Our framework — modeling language and scope

Maude: a formal language supporting

- ✓ Expressive formalism based on ***rewriting logic***
- ✓ ***Concurrent*** computation with ***state***
- ✓ ***Extensive*** tools for formal specification & verification



Our scope: ***end-to-end*** name resolution up to the ***latest*** algorithmic refinements in RFC9156

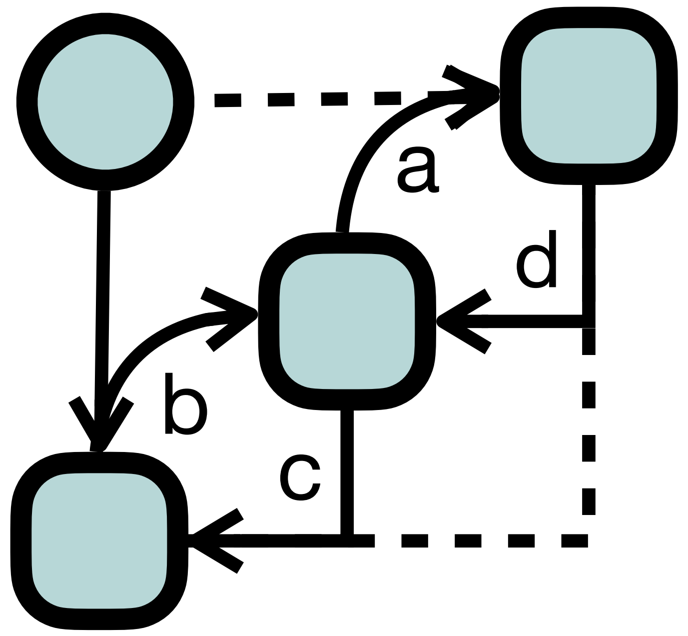


Abstracted away in GRoot [Sigcomm'20]

Our model's RFC coverage

RFC	Description
1034 [36]	Core specification
1035 [37]	Core specification
2181 [21]	Clarifications
2308 [6]	Negative caching
4592 [29]	Wildcards
6604 [25]	RCODE clarifications
6672 [40]	DNAME redirection
8020 [12]	NXDOMAIN clarification
9156 [11]	QNAME minimization

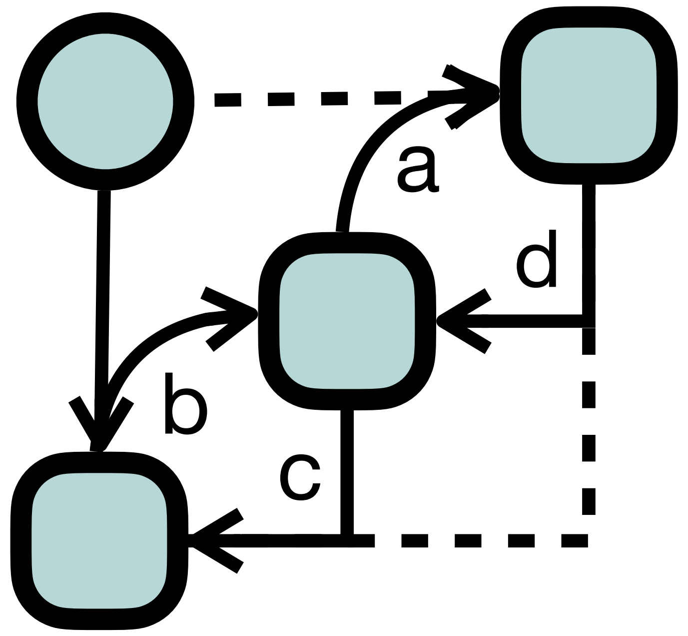
Our framework — executable DNS semantics



Modeled as labelled transition system in ***actor paradigm***

- System dynamics specified by rewriting rules
- Non-deterministic and probabilistic variants

Our framework — executable DNS semantics



Modeled as labelled transition system in ***actor paradigm***

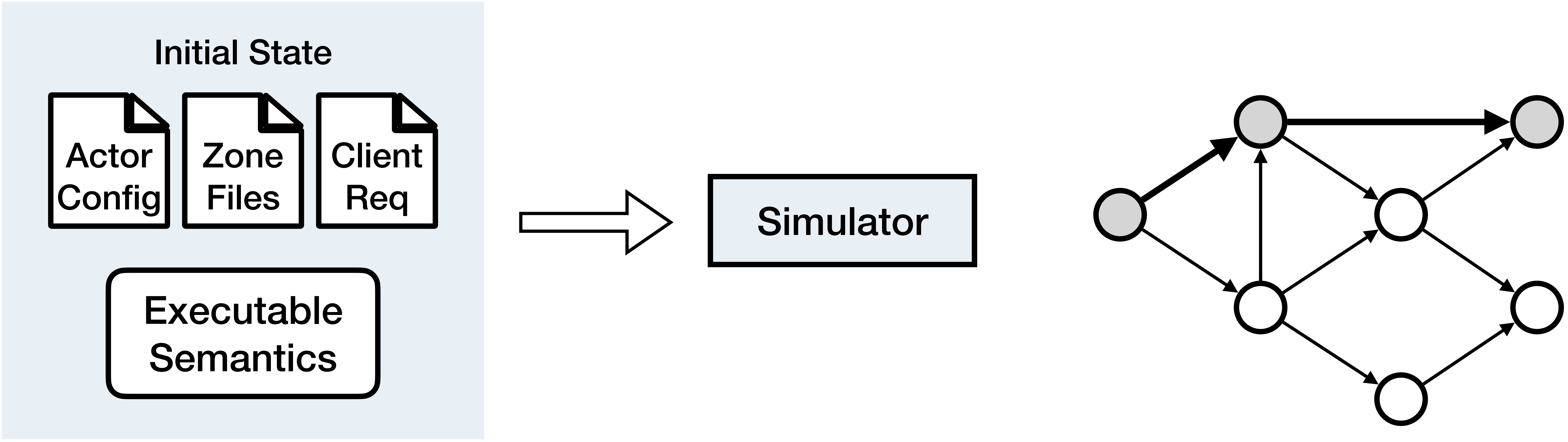
- System dynamics specified by rewriting rules
- Non-deterministic and probabilistic variants

Resolve ambiguities whenever possible, e.g., resolver case distinction; otherwise, make them ***configurable***, e.g., data credibility rule

Option	Definition	Default
`rsvMinCredClient`	The minimum credibility requirement [21] for data served to a client	2
`rsvMinCredResolver`	The equivalent credibility requirement for resolver subqueries	2
`maxMinimiseCount`	The MAX_MINIMIZE_COUNT parameter to limit extra work for QMIN [11]	10
`minimiseOneLab`	The MINIMIZE_ONE_LAB parameter from the same mechanism above	4
`rsvTimeout`	Whether and how long a resolver applies a timeout for each query it sends	false, 20.0
`rsvOverallTimeout`	Whether and how long a resolver applies an overall timeout for a client request	false, 100.0

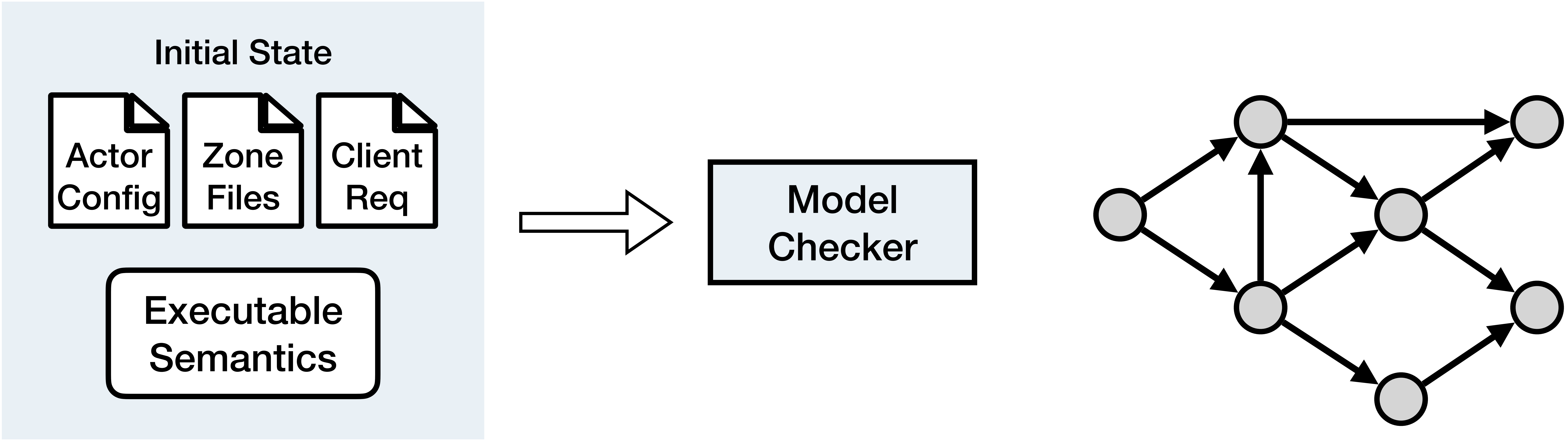
Our framework — formal analysis

Simulation for semantics sanity checks, serving as reference implementation



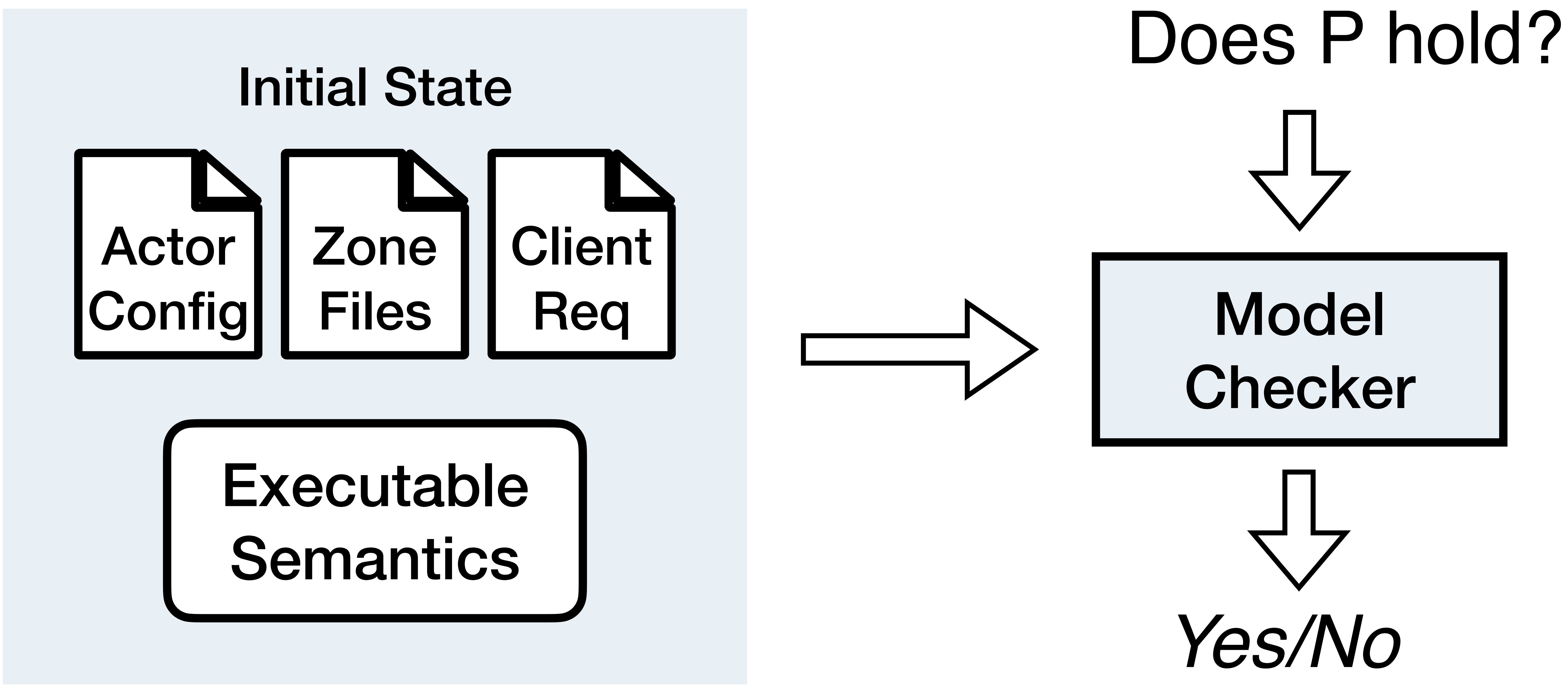
Our framework — formal analysis

Model checking on ***qualitative*** properties



Our framework — formal analysis

Model checking on ***qualitative*** properties, e.g., RFC compliance of zone config



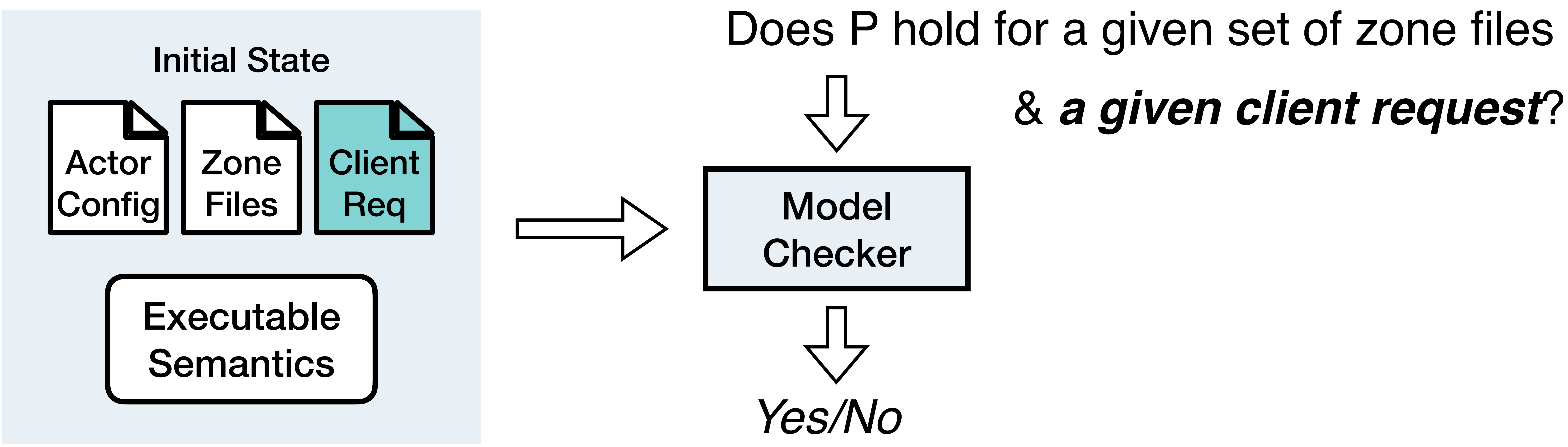
P := absence of

- lame delegation
- circular dependency
- answer inconsistency
- rewrite blackhole
- …

Include all properties in GRoot [Sigcomm'20]

Our framework — formal analysis

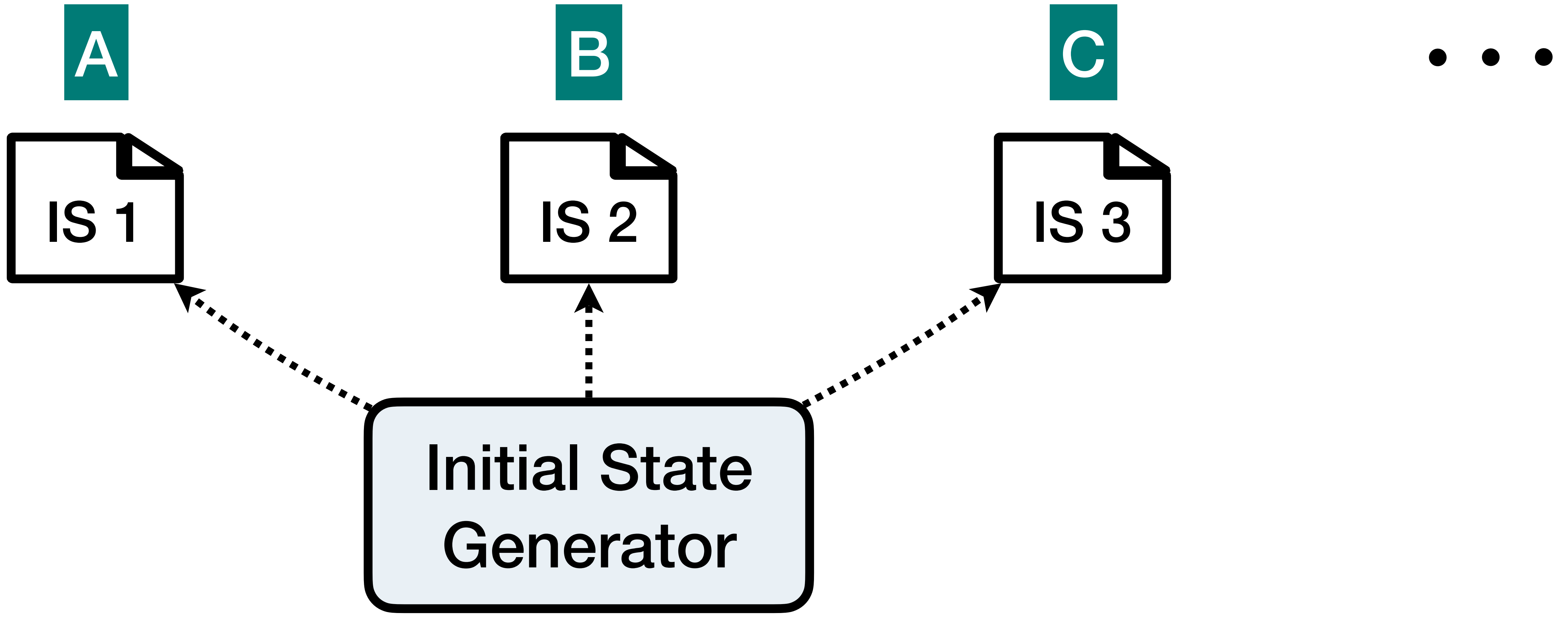
Model checking on ***qualitative*** properties, e.g., RFC compliance of zone config



Our framework — formal analysis

Model checking on ***qualitative*** properties, e.g., RFC compliance of zone config

Non-deterministic ***initial state exploration*** with automation

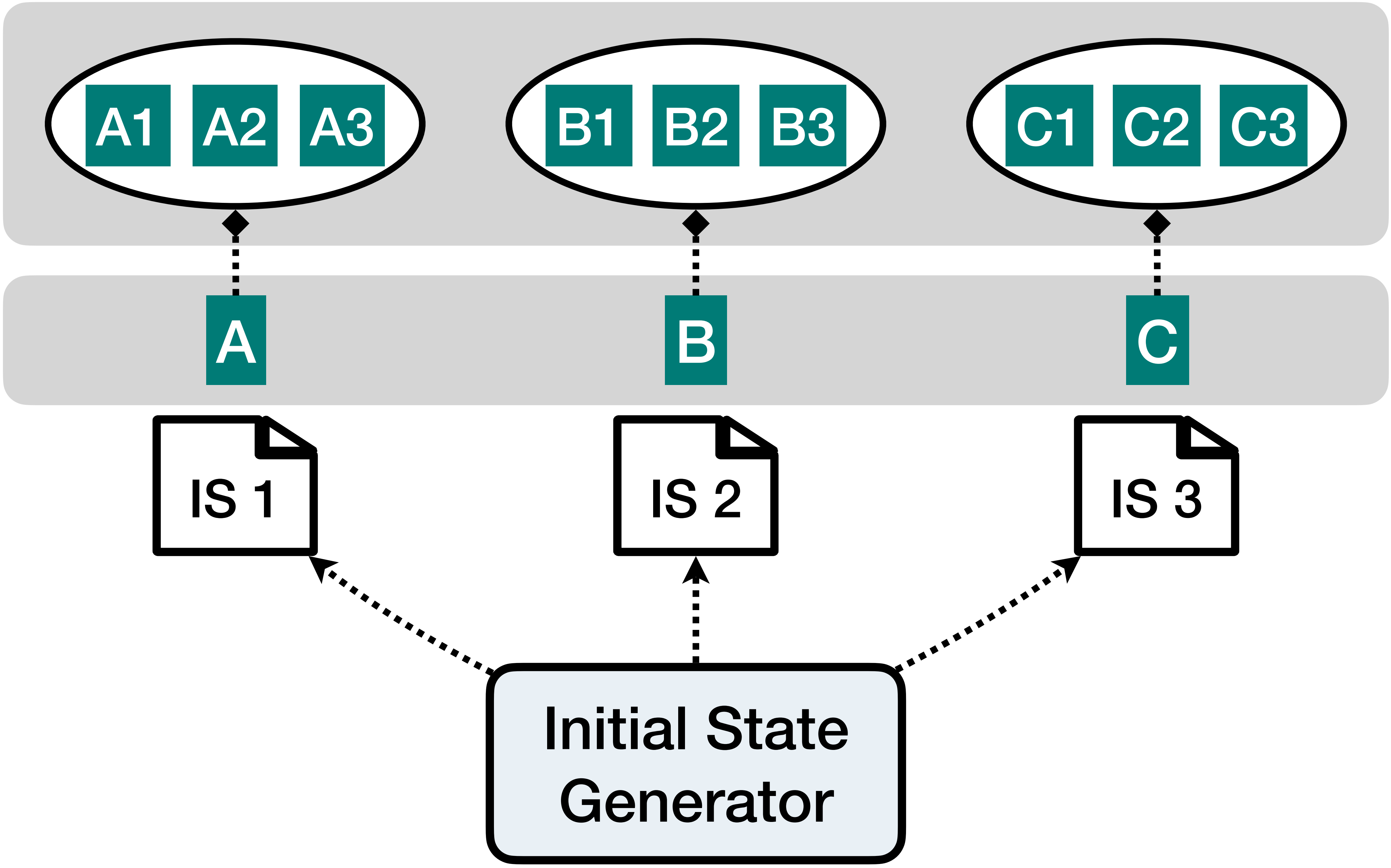


query space is huge!

Our framework — formal analysis

Model checking on ***qualitative*** properties, e.g., RFC compliance of zone config

Non-deterministic ***initial state exploration*** with automation



Query equivalence class (EC)

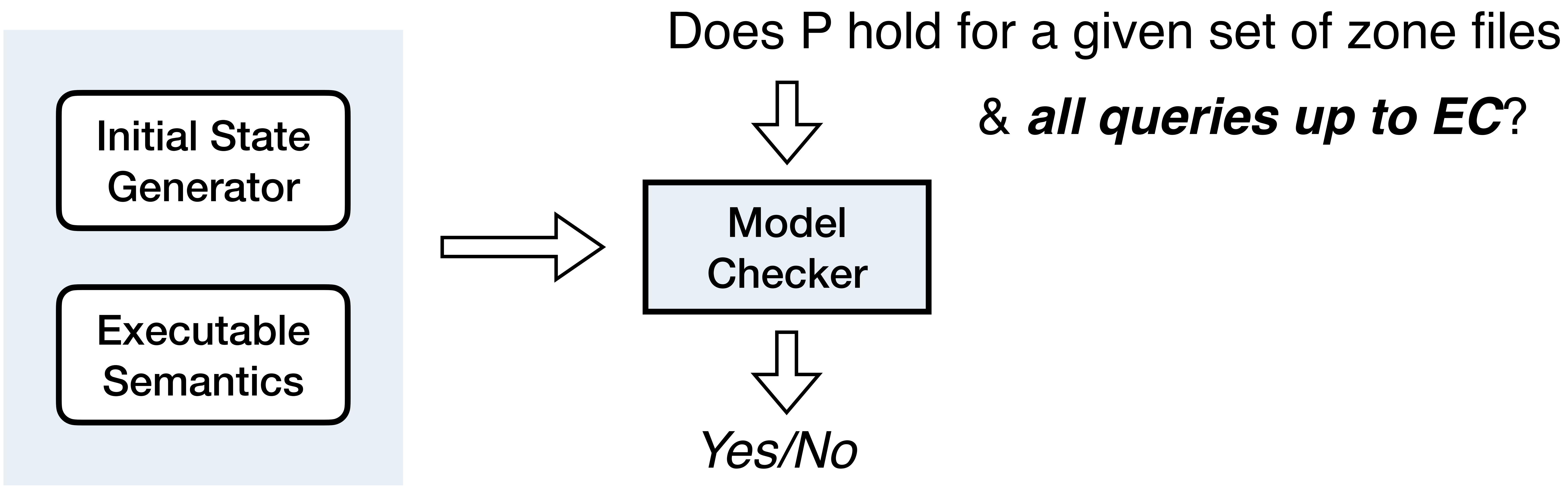
Sample query from each EC

Caveat: definition of EC is critical

- Use GRoot's EC as a ***heuristic***

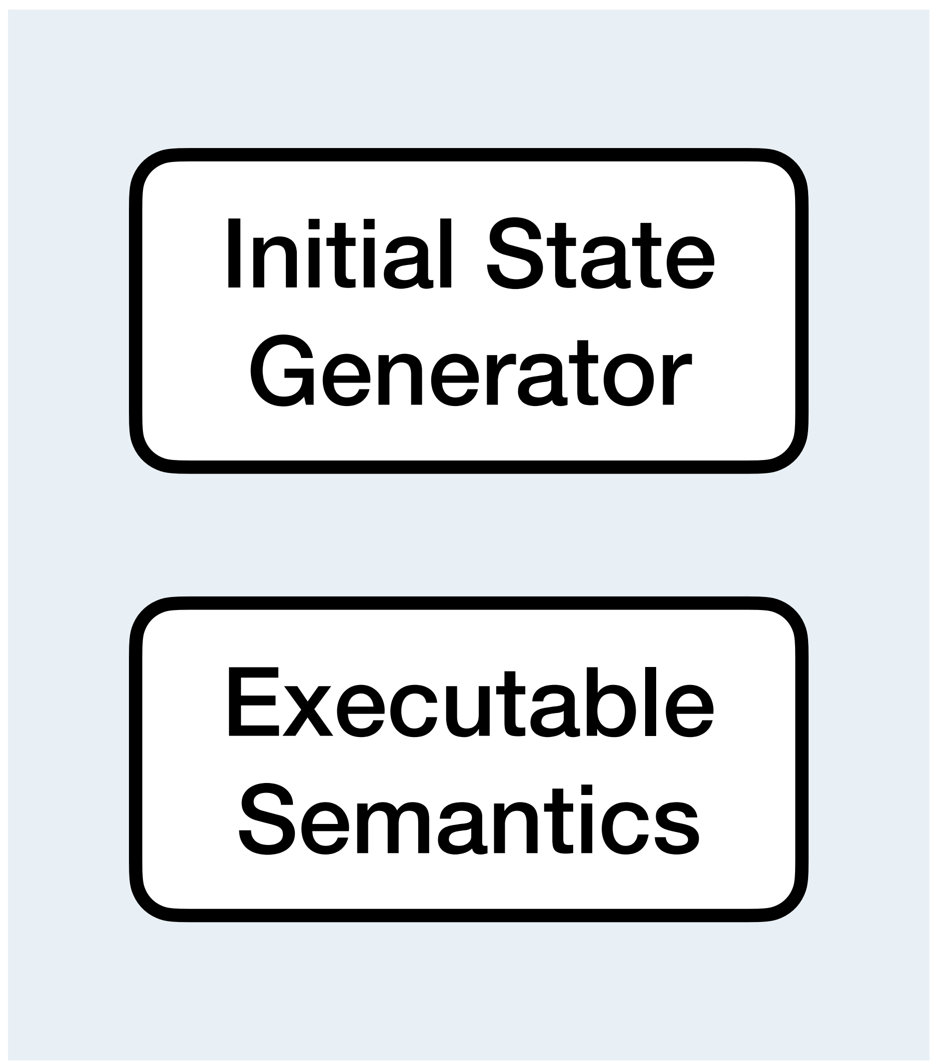
Our framework — formal analysis

Model checking on ***qualitative*** properties, e.g., RFC compliance of zone config

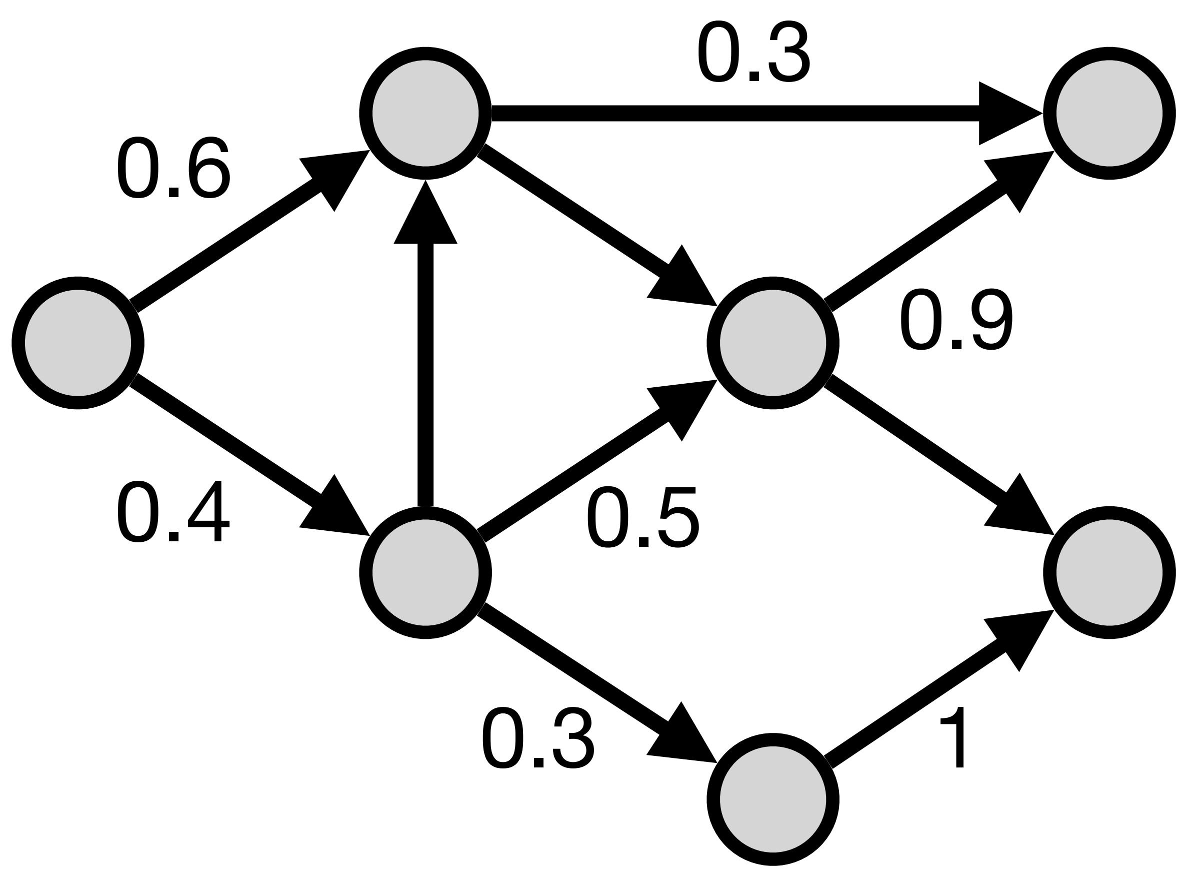


Our framework — formal analysis

Statistical verification on ***quantitative*** properties

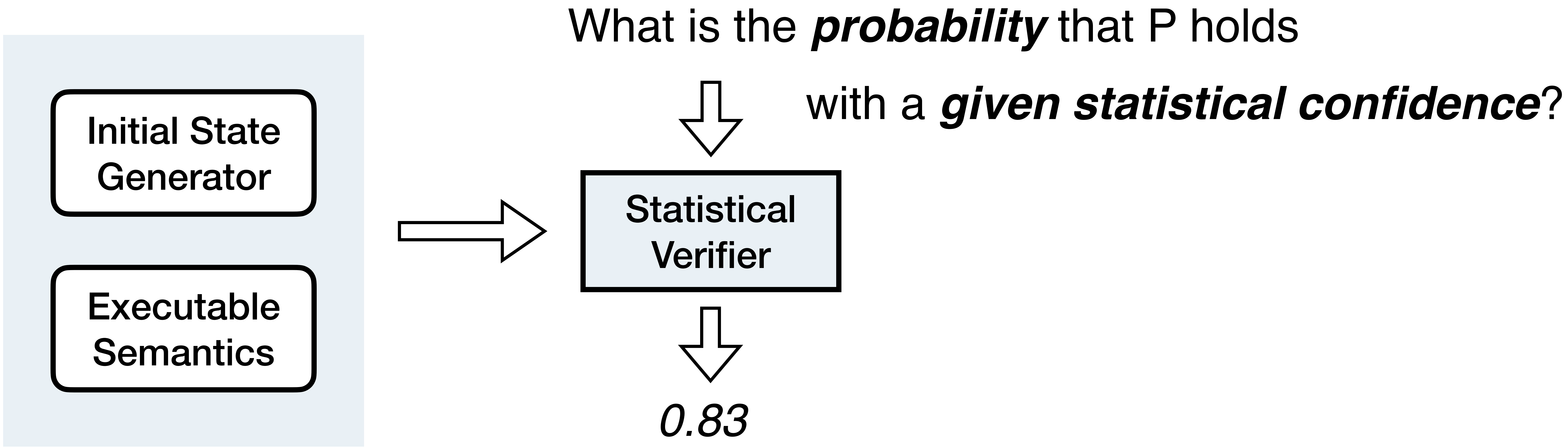


E.g., PVeStA SMC



Our framework — formal analysis

Statistical verification on ***quantitative*** properties, e.g., query success ratio



Our framework — formal analysis

Statistical verification on ***quantitative*** properties, e.g., query success ratio

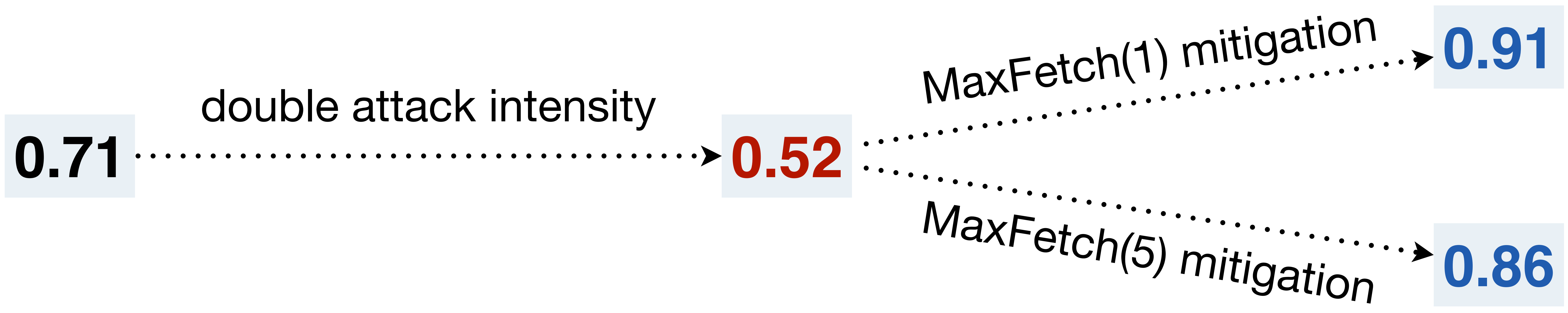
Example: Under NXNS attack [SEC'20], with 0.05 error margin and 95% statistical confidence, the query success ratio of a legitimate client is

0.71

Our framework — formal analysis

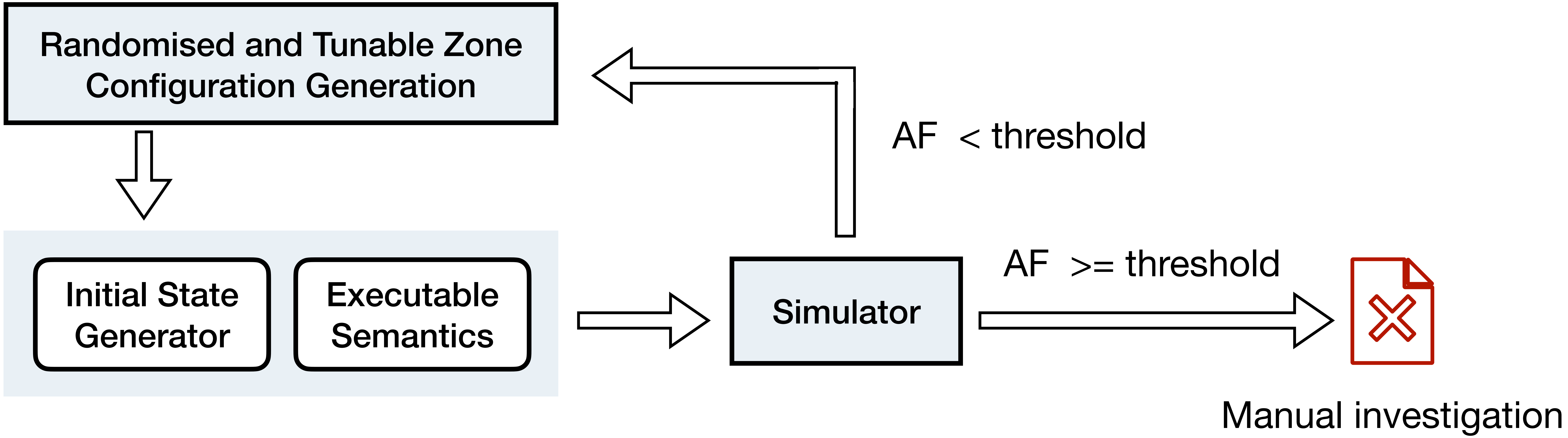
Statistical verification on ***quantitative*** properties, e.g., query success ratio

Example: Under NXNS attack [SEC'20], with 0.05 error margin and 95% statistical confidence, the query success ratio of a legitimate client is



Application: automated analysis of DoS vulnerabilities

Excessive queries triggered by a single client request: high ***amplification*** factor (AF)



Application: automated analysis of DoS vulnerabilities

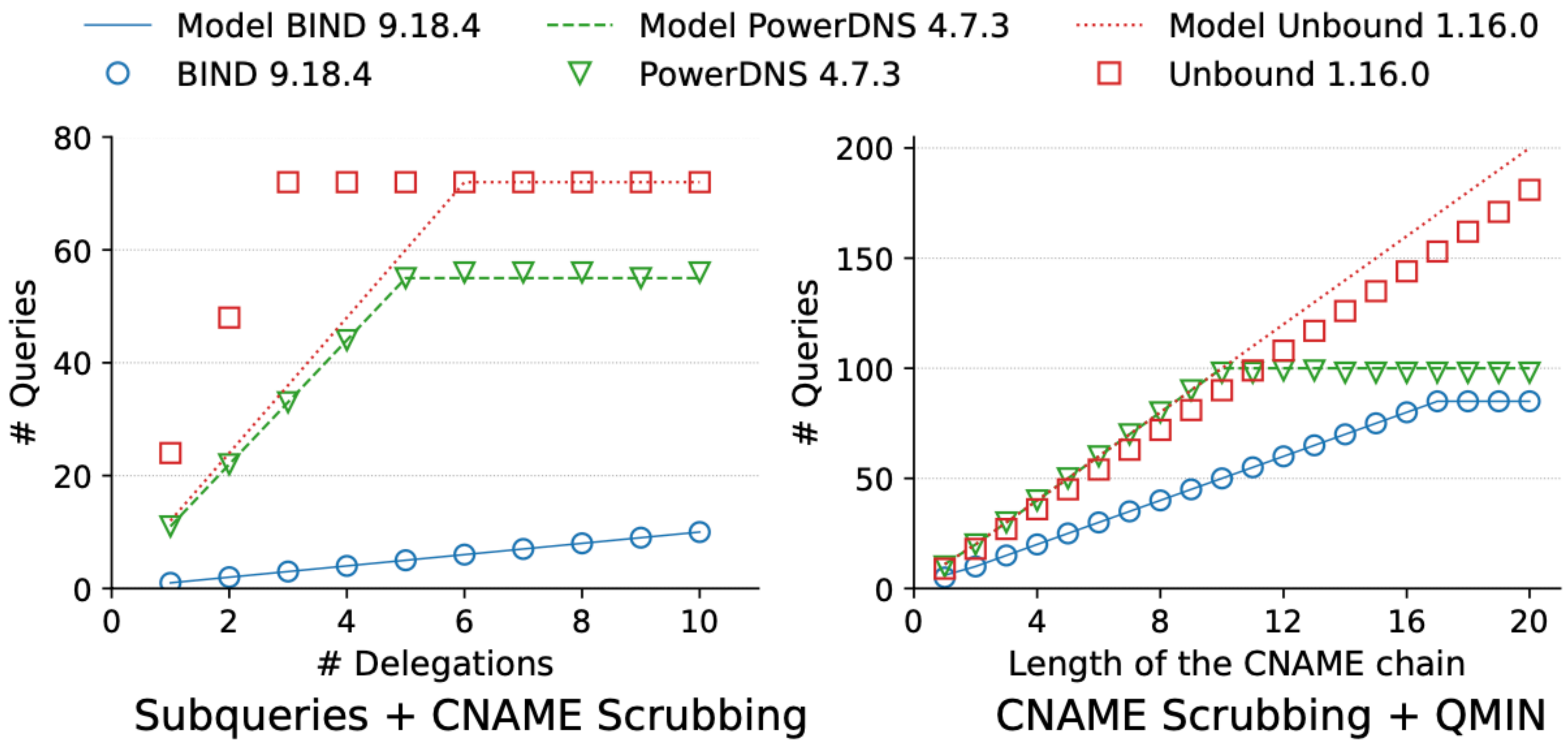
Excessive queries triggered by a single client request: high ***amplification*** factor (AF)

Re-discovered major known vulnerabilities [DNS-OARC'15, RAID'18, SEC'20, IMC'21]

New vulnerabilities

- Exploit interaction btwn features
- 100s of MAF
- Validated in DNS software
- Reported, investigation WIP

See paper for detail!



Summary and outlook

Our framework establishes a formal foundation for DNS

- ***Comprehensive*** semantics
- ***Versatile*** in verification (quantitative property 1st time)
- ***Automated*** toolset

Give it a try

Summary and outlook

Our framework establishes a formal foundation for DNS

- ***Comprehensive*** semantics
- ***Versatile*** in verification (quantitative property 1st time)
- ***Automated*** toolset

Give it a try

Future work

- More DNS features, e.g., DNSSEC, DoT/DoH
- Richer property library, better automation
- Sound and complete definitions of EC

Thank you!

Questions?

Contact: huayi.duan@inf.ethz.ch, si.liu@inf.ethz.ch