University of Leicester

Detecting and Refactoring Operational Smells within the Domain Name System

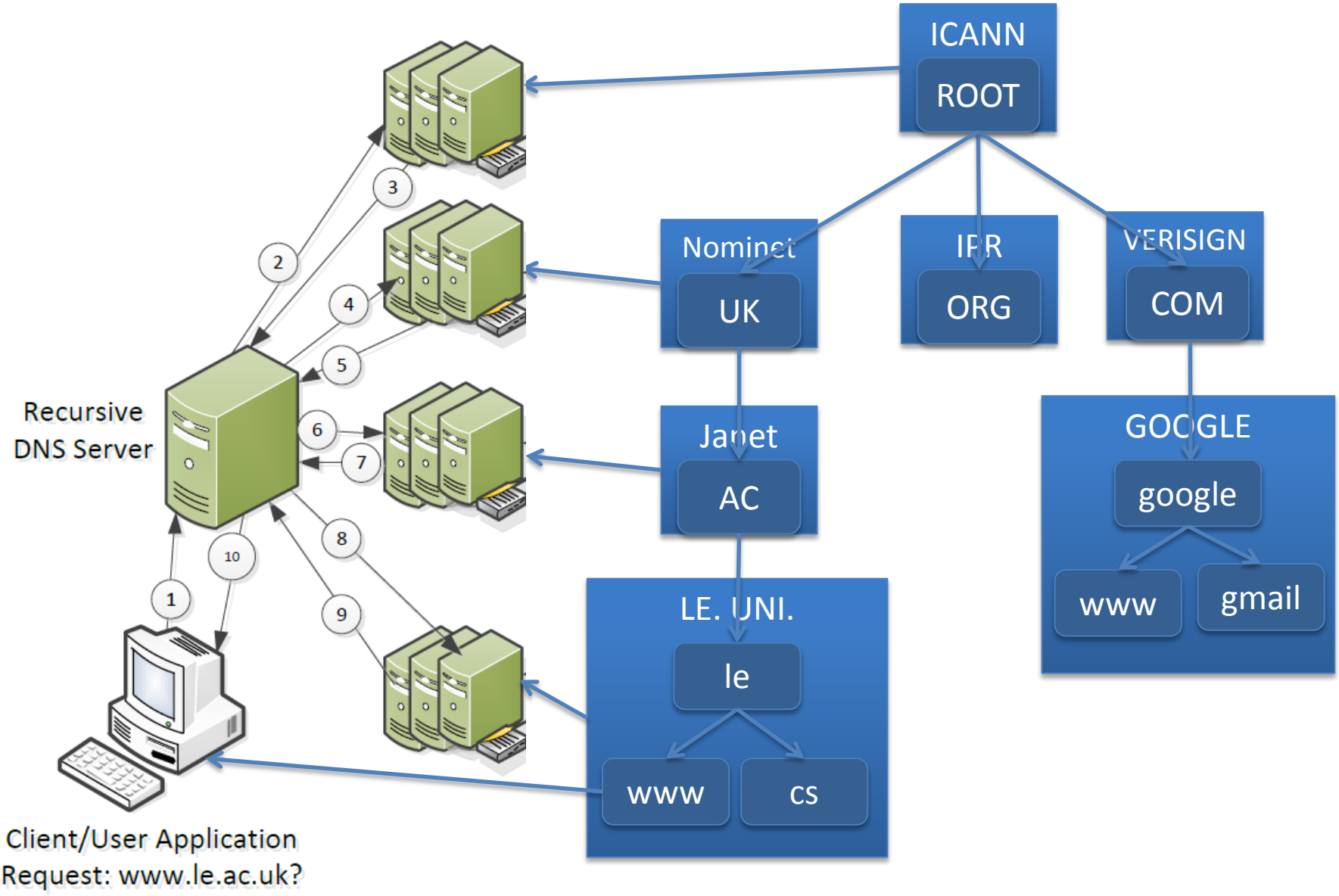
Graphs as Models (GaM) Workshop,
European Joint Conferences on Theory and Practice of Software (ETAPS-15)
11,12 April 2015
Queen Mary University, London, United Kingdom

Marwan Radwan and Reiko Heckel

Outline

1. The Domain Name System
2. Problem Statement
3. DNS Dependency Graphs and Operational Model
4. ISDR Method
5. Refactoring and Graph Transformations
6. Challenges and Future Work
7. Q&A

The Domain Name System (DNS)

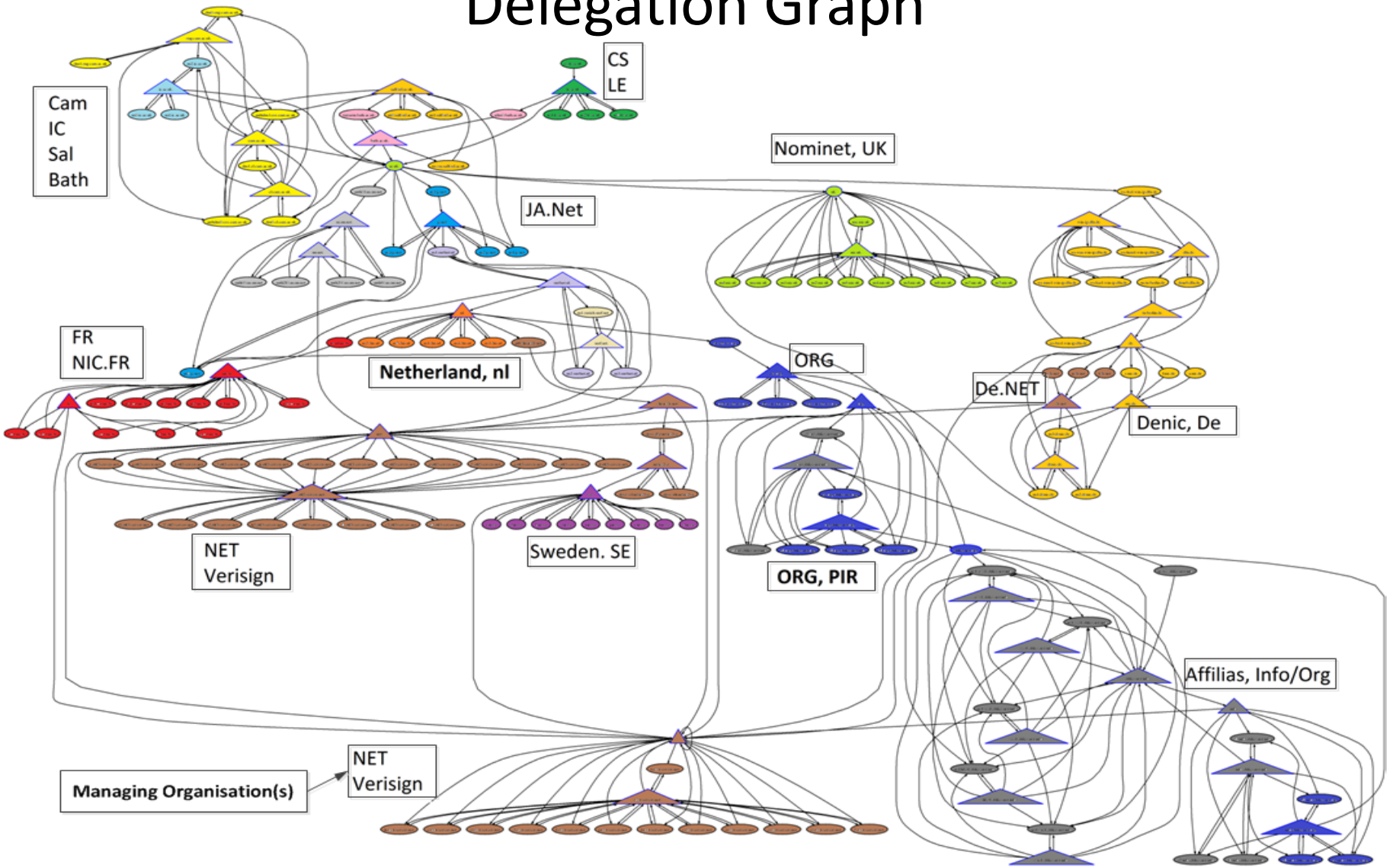


The Problem:

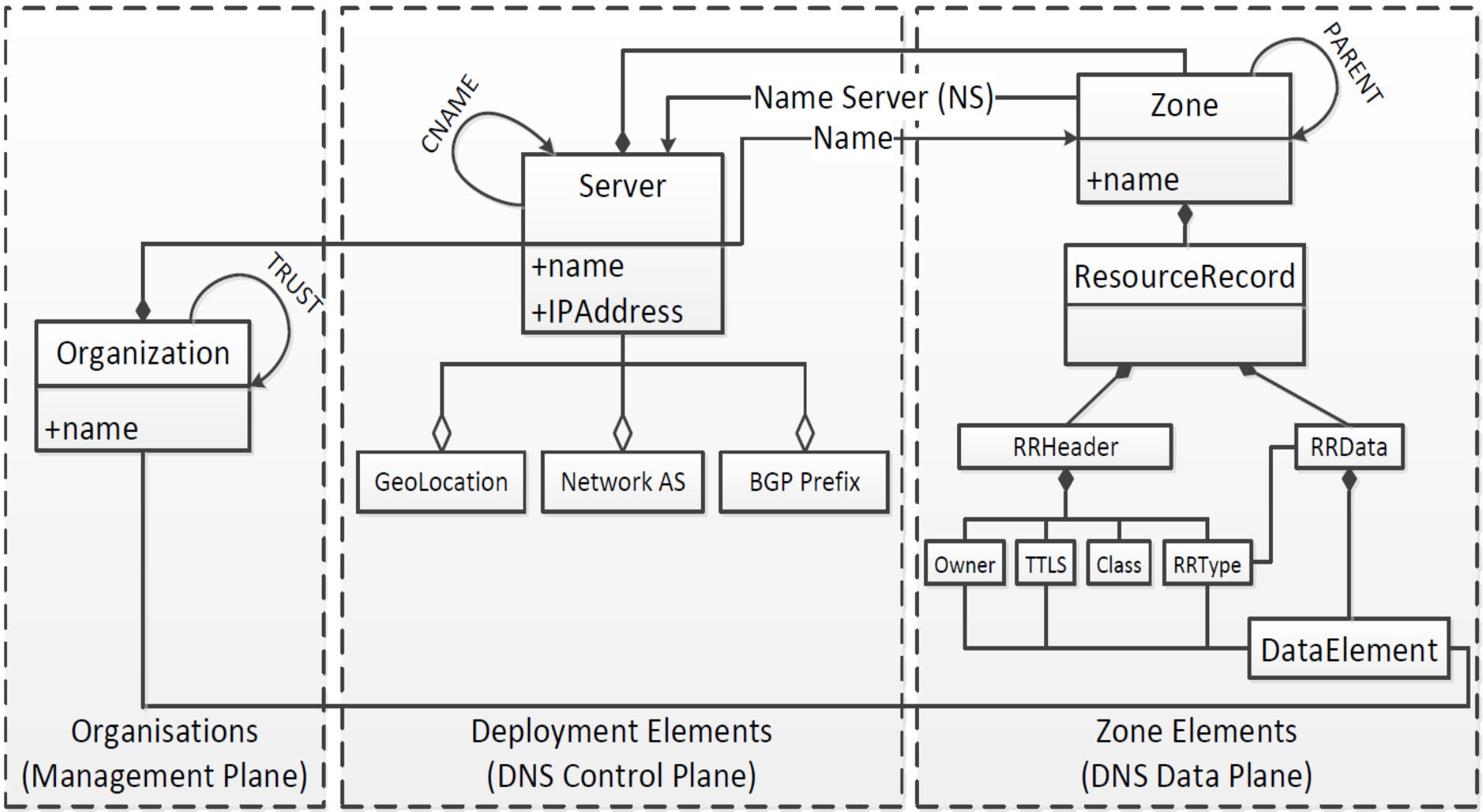
Operational Parameters & Implications on DNS System's Qualities

- DNS Operational Interdependencies
  - System Administrator Actions: Zone Misconfiguration and Name Servers' Deployment Choices
  - Quality Impacts (Availability, Security and Resilience)
- Examples:
  - Servers' names : Zone redundancy ***but*** security and resiliency threats (Zone Influence, Attack Surface and Points of Failure).
  - # of ANSs and where to locate them: Server Redundancy ***but*** operational overhead and less resistance against failures.
  - Peering with external organizations: Diversity ***but*** transitional trust, politics and administrative complexity.

DNS Interdependencies
Delegation Graph



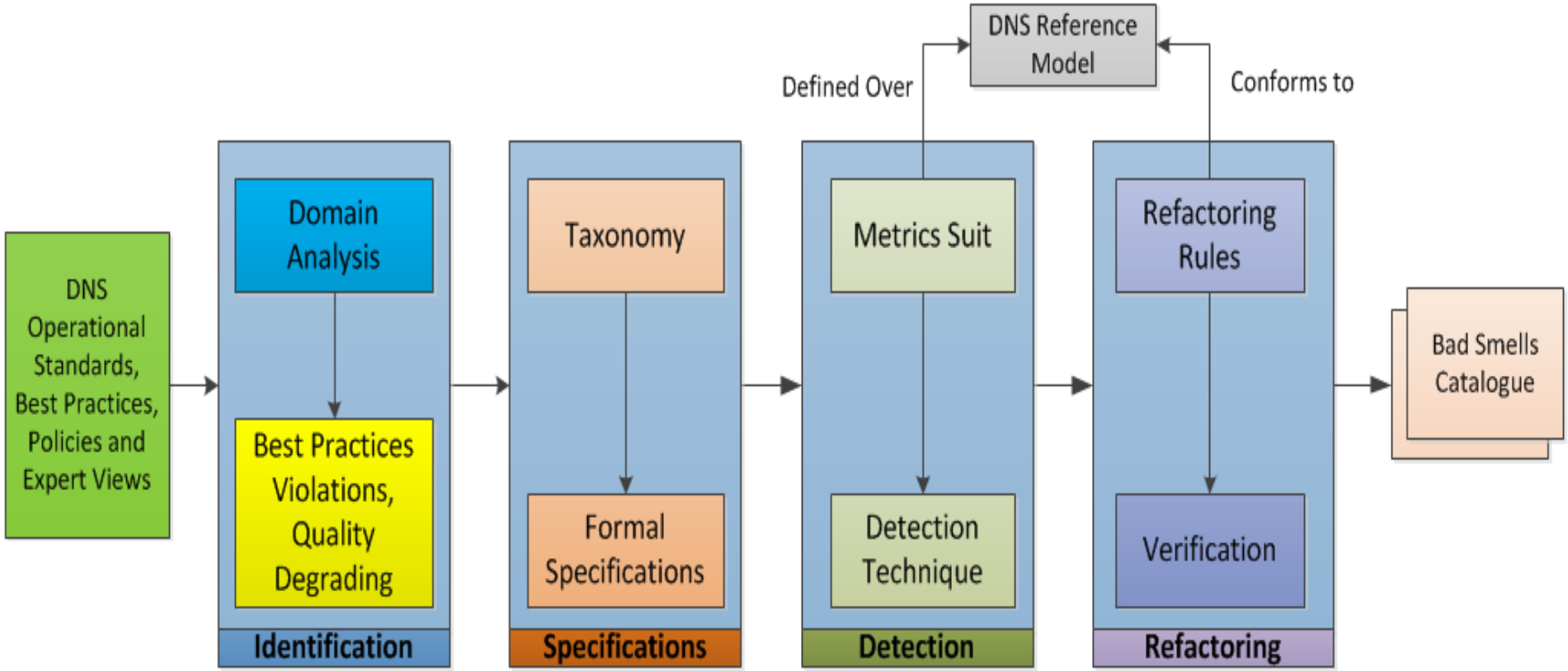
DNS Operational Model



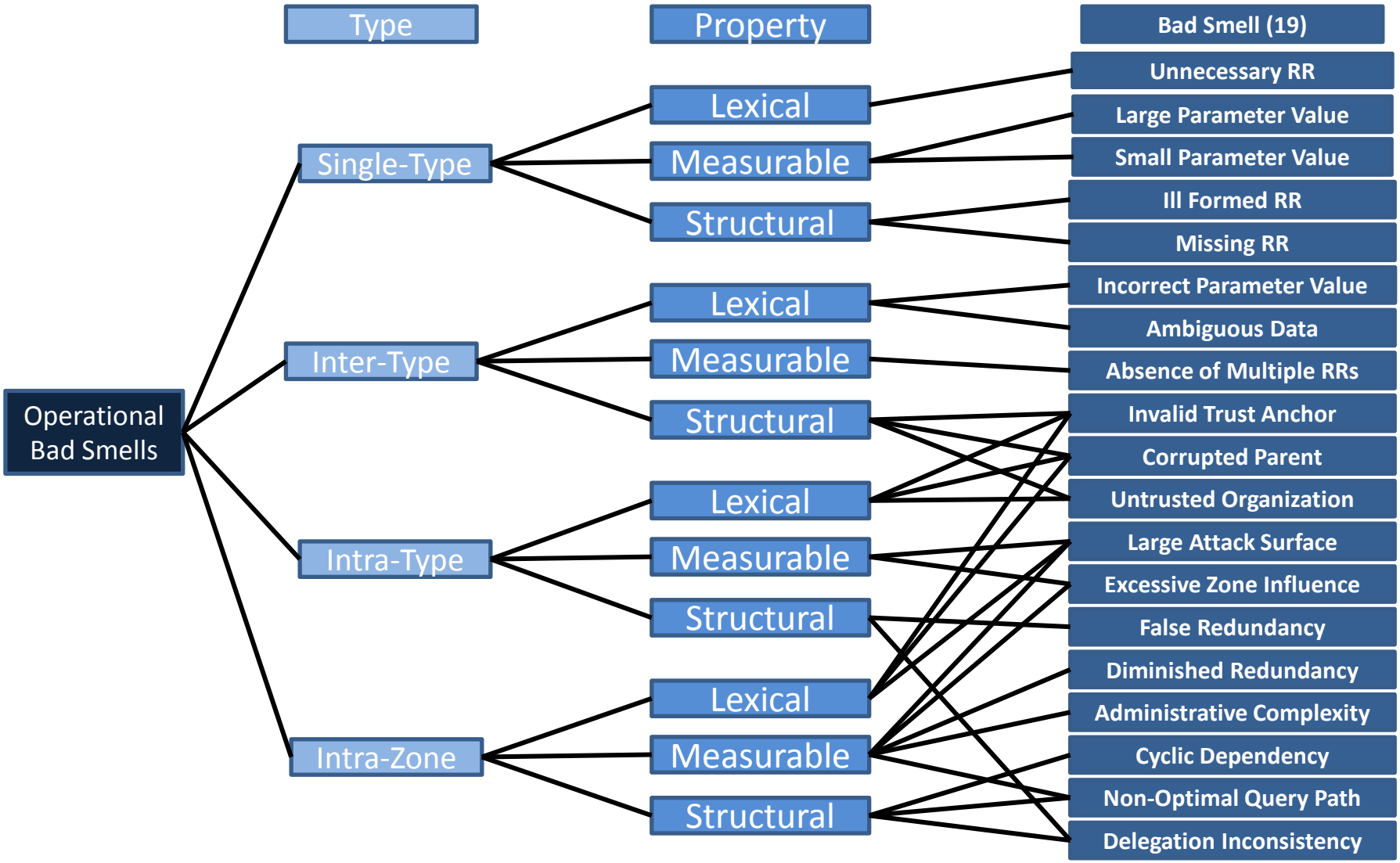
Operational Entities, Their Properties and Relations

The ISDR Method

Model-based approach that subsumes all the steps necessary to identify, specify and detect the DNS operational bad smells.



Bad Smells Taxonomy



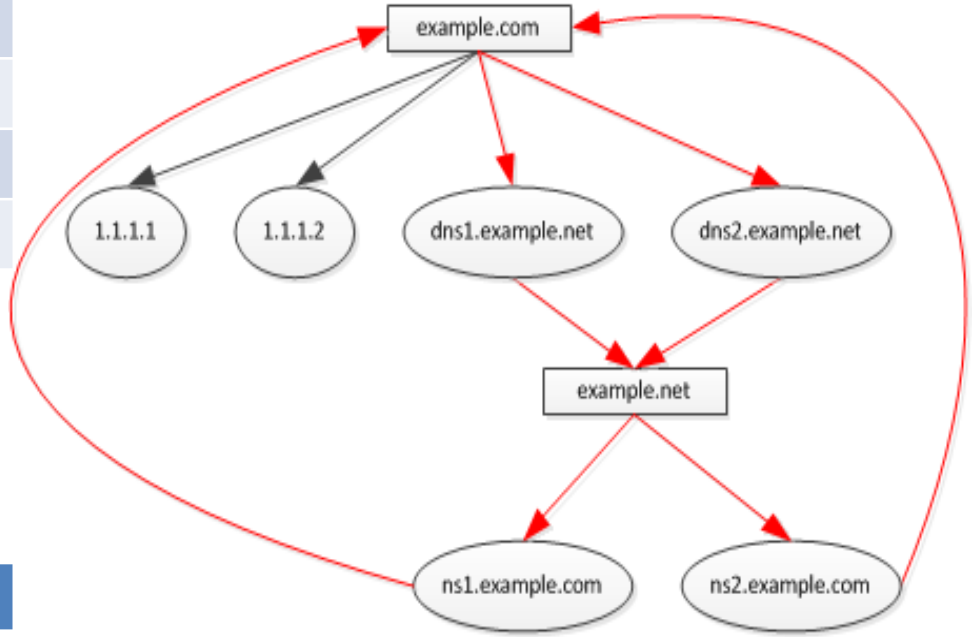
DNS Metrics

Metric	Administrative Complexity
Definition	Describes the diversity of a zone with respect to the organisations administering its authoritative name servers.
Usage	Mutual hosting of zones between organizations is common in the DNS deployment schemes. The advantage of such practice is an increased availability but at the same time increased potential of failure and instability of the DNS zone resolution process.
How to Measure	Count the number of authoritative name servers of each organization involved in the dependency graph of zone z.
Metric Notation	O_z denotes the set of organizations that administer authoritative name servers hosting the zone *z*; *n* denotes total number of authoritative name servers; NS^o_z denotes subset of name servers administered by organization o in O_z.
Formula	$1 - \sum_{o}^{n} \left(\frac{NS_z^o}{NS_z}\right)^n$

Validation
Case Study-1: Cyclic Dependency

$ORIGIN .com. [Parent of example.com]		
example.com.	NS	ns1.example.com.
example.com.	NS	ns2.example.com.
example.com.	NS	dns1.example.net.
example.com.	NS	dns2.example.net.
ns1.example.com.	A	1.1.1.1
ns2.example.com.	A	1.1.1.2

$ORIGIN .net. [Parent of example.net]		
example.net.	NS	ns1.example.com.
example.net.	NS	ns2.example.com.



Refactoring: *AddGlueRecord*

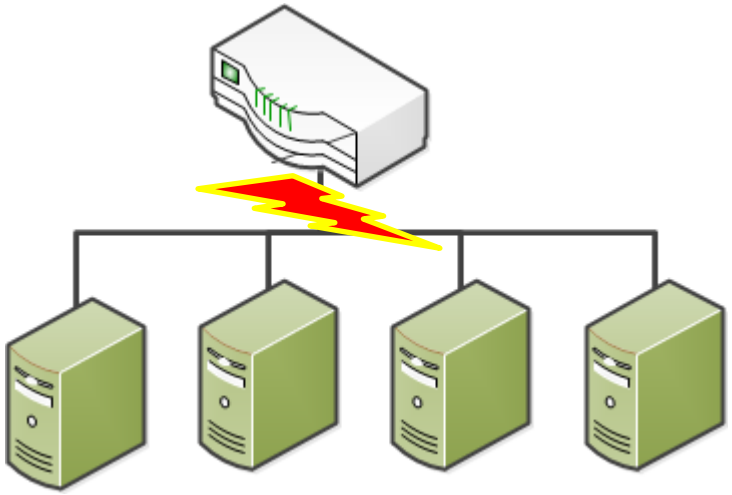
- Same Administrative Unit (Example Company)
- Different Zones.

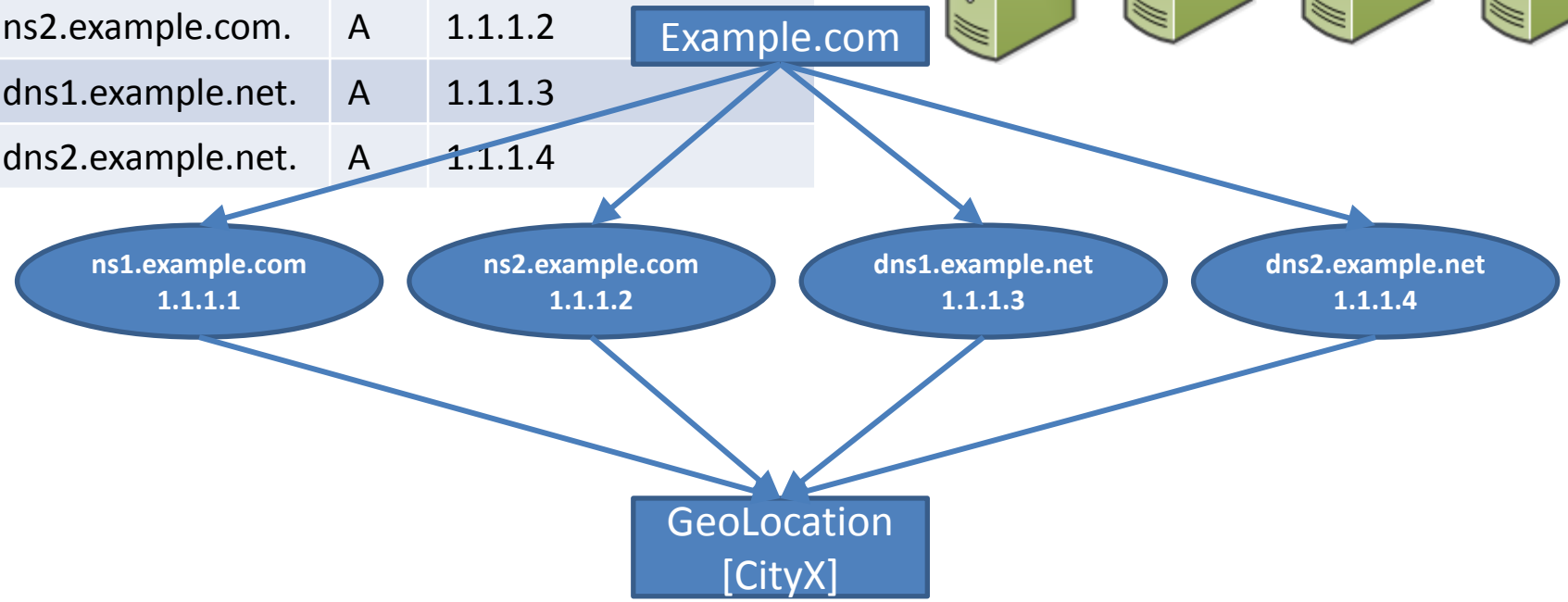
dns1.example.net.	A	1.1.1.3
dns2.example.net.	A	1.1.1.4

Detection: Is there any cycle in the DG?

Case Study-2: False Redundancy

$ORIGIN .com.		
example.com.	NS	ns1.example.com.
example.com.	NS	ns2.example.com.
example.com.	NS	dns1.example.net.
example.com.	NS	dns2.example.net.
ns1.example.com.	A	1.1.1.1
ns2.example.com.	A	1.1.1.2
dns1.example.net.	A	1.1.1.3
dns2.example.net.	A	1.1.1.4





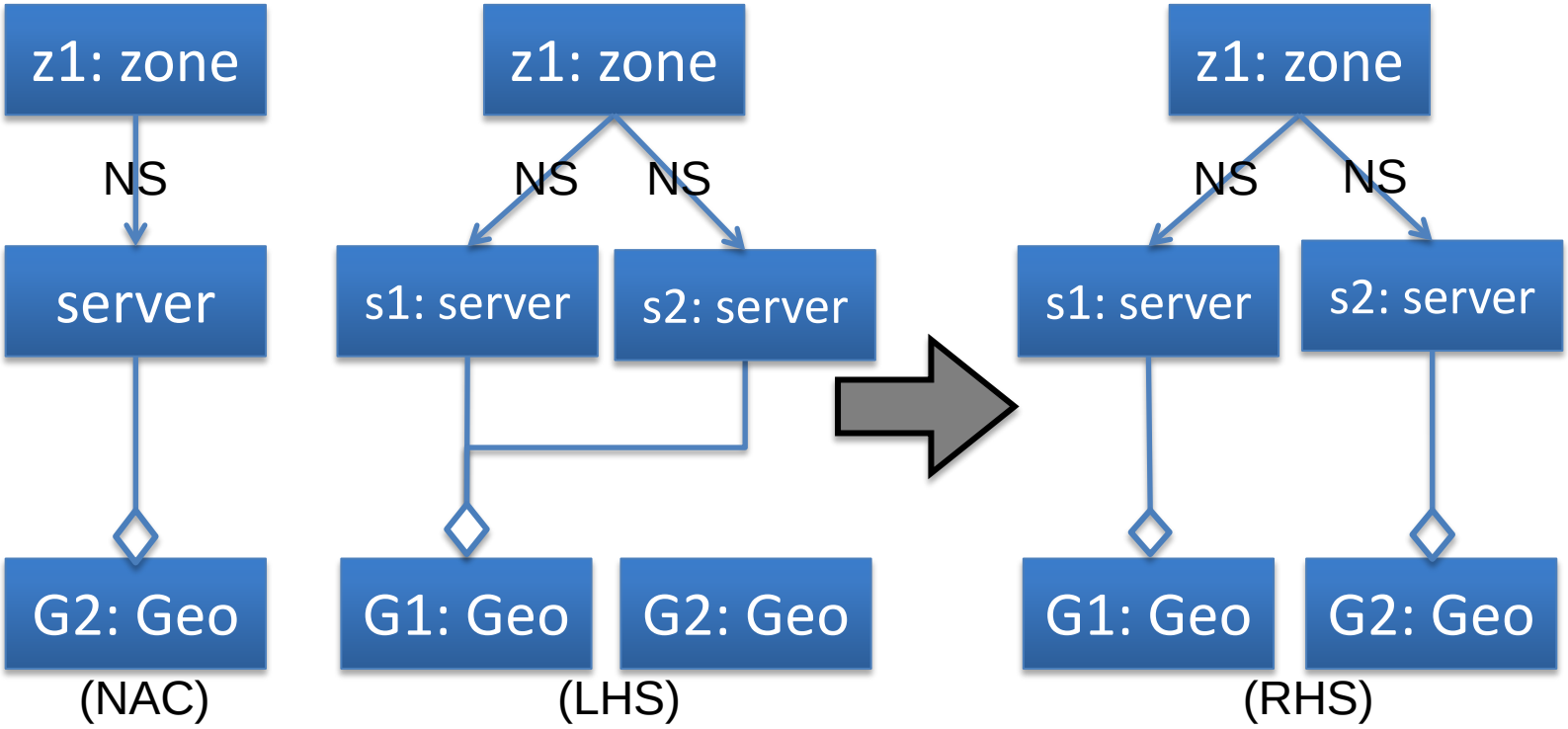
Bad Smell Catalogue Entry

Name	False-Redundancy.
Type	Inter-zone and Measurable.
Inspection Planes	Control Plane.
Description	Level of availability of redundant servers function of their number, physical location and networks they connect to.
Occurrence	All redundant servers at the same physical location, connected to the same network
Quality Impacts	Reduced availability & resilience, and the system become susceptible to single point of failure at certain granularity.
Detection Strategy	Queries on DG for the following metrics: a) # of ANSs, b) # of GeoLocations, c) # of ASs, and d) distinct BGP prefixes.
Refactoring	Applying MoveServerLocation refactoring rule will improve the availability of the zone and its resilience.

Addressing Bad Smells Through GT-Based Refactoring

- **Refactoring** is the technique of choice for ***improving*** the ***structure*** of existing code ***without changing*** its ***external behaviour***.
- Refactoring implementation includes the following steps:
  1. Identify the location for refactoring,
  2. Refactoring Rules priorities and conflicts,
  3. Guarantee the preservation of external behaviour of the system,
  4. Application of selected refactoring rules,
  5. Assess the effect of refactoring on the systems external qualities and
  6. Maintain the consistency between the refactored elements and other system artefacts.

Refactoring Rule: *MoveServerLocation*



NAC: No Servers have already been located in the G2 GeoLocation
LHS: Two ANSs are located within the same GeoLocation (G1)
RHS: One of the Servers is moved to the new GeoLocation (G2)
Quality Impact: Improved Availability and Resilience against G1 “Failures”

Execution of Refactoring Rules

- Priorities and Alternative Rules
  - MoveServerLocation Vs Creating a new server at the new GeoLocation.
  - Quality Impacts, Trade-offs on NFP and cost.
  - More than one rule application, incremental improvement, repeated or combined with others.
- Self-Contained Refactoring Rules
  - Executed by directly modifying the zone file records elements within the model.
- Refactoring Across Borders
  - Physical and Access Permissions.
  - Coordination with other zones' administrators and peer organizations (SLA).

Future Work and Challenges

- Bad Smells Catalogue
  - Expand the Taxonomy,
  - DNS Metrics Suite, and
  - Detection Logical Formulas and Techniques.
- Refactoring Rules:
  - Verification: behaviour preservation and quality improvements.
  - Applicability: trade-offs, conflicts and ordering.
  - Complex sequence of operations to transform the model changes into physical resources relocation.
  - Physical access permissions, or coordination actions such as service level agreements (SLAs) with external sites/organizations.

Future Work and Challenges

- DNS Operational Advisor (DOA) Tool
  - DG extraction from Zone file(s) and Deployment Choice(s)
  - DNS Qualities (Availability, Security and Resiliency)
  - DNS Structural Metrics
  - New configuration generation and DNS operator recommendations
  - Semi-Automated DNS Quality Dashboard and/or Reports