Astrolabe:

A Robust and Scalable Technology for Distributed System

R. van Renesse, K.P. Birman, W. Vogels *(Cornell University)*

Presentation by Konrad Iwanicki

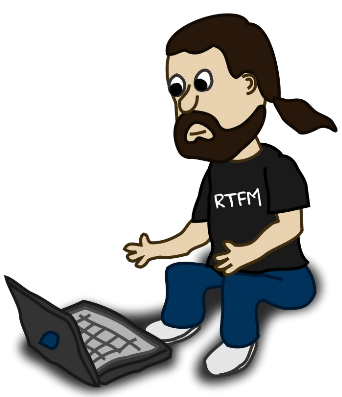
Introduction

- Imagine an organization that needs to manage large collections of distributed resources, such as:
  - personal workstations
  - dedicated nodes in a web farm
  - or objects stored and services run on these computers.
- Think really big:
  - Amazon.com,
  - Google.com,
  - University of Warsaw,
  - Your own Internet start-up.

Sample Objects & Resources

pc372.mimuw.edu.pl

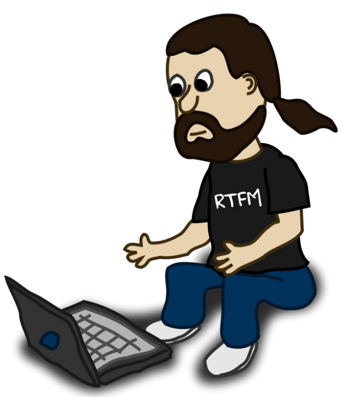
Name	pc372
Browser	IE
Printer	OKI 783
Disk_total	20GB
Disk_used	5GB



laptop065.mimuw.edu.pl

Name	laptop065
Browser	Firefox
Printer	-
Disk_total	500GB
Disk_used	413GB

Sample Management



Name	pc372	duch	laptop065
Browser	IE	-	Firefox
Printer	OKI 783	HP 3971	-
Disk_total	20GB	2000GB	500GB
Disk_used	5GB	587GB	413GB

laptop065.mimuw.edu.pl

Problems

- The computers hosting the resources may be:
  - co-located in a room,
  - spread across a building or a campus,
  - scattered around the world.
- Configurations of such systems change rapidly:
  - machine failures and connectivity changes are common,
  - significant adaptation may be necessary to provide the desired level of service.
- How do you manage the resources in such systems?
  - In particular, how do you retrieve information about the resources?

Sample Management Queries

SELECT

COUNT(user_name) AS firefox_user_count

FROM processes

WHERE

name = 'firefox.exe'

GROUP BY

name

SELECT

FIRST(3, host_name) AS host

WHERE

Disk_used/Disk_total > 80%

ORDER BY

Disk_used/Disk_total DESC

Requirements

- Resource information aggregation:
  - A convenient way of getting some summaries regarding the resources in the system.
- Resource location:
  - A means for locating resources based on the summaries.

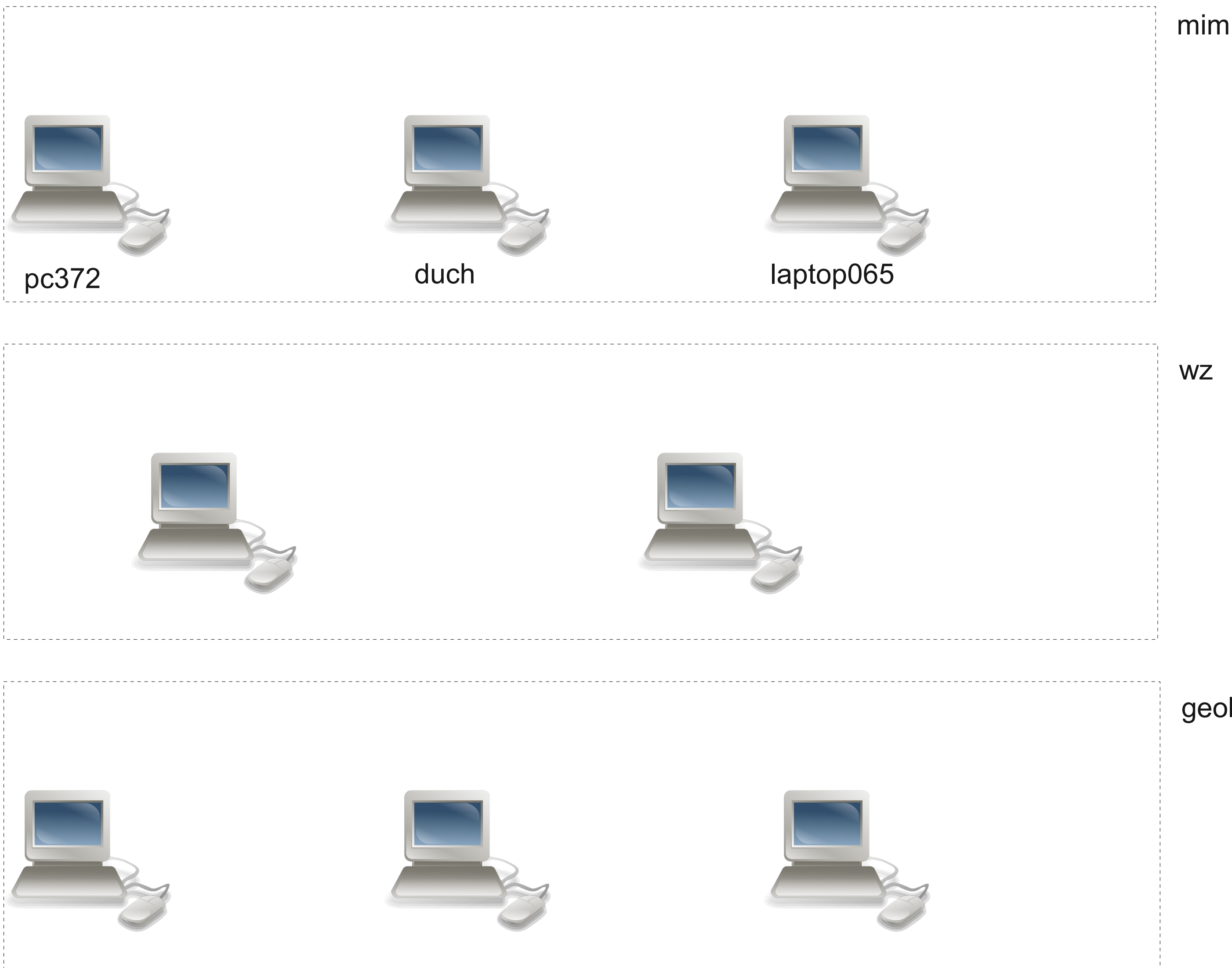
- Scalability (in terms of the number of machines).
- Robustness to changes in the network topology.
- Security (access control and integrity).

A solution: Astrolabe

- Developed at Cornell University and a start-up company, RNS.
- Used by Amazon.com to manage its huge collections of machines at the services those machines run.
  - Werner Vogels is now the CTO of Amazon.com (http://www.allthingsdistributed.com/)

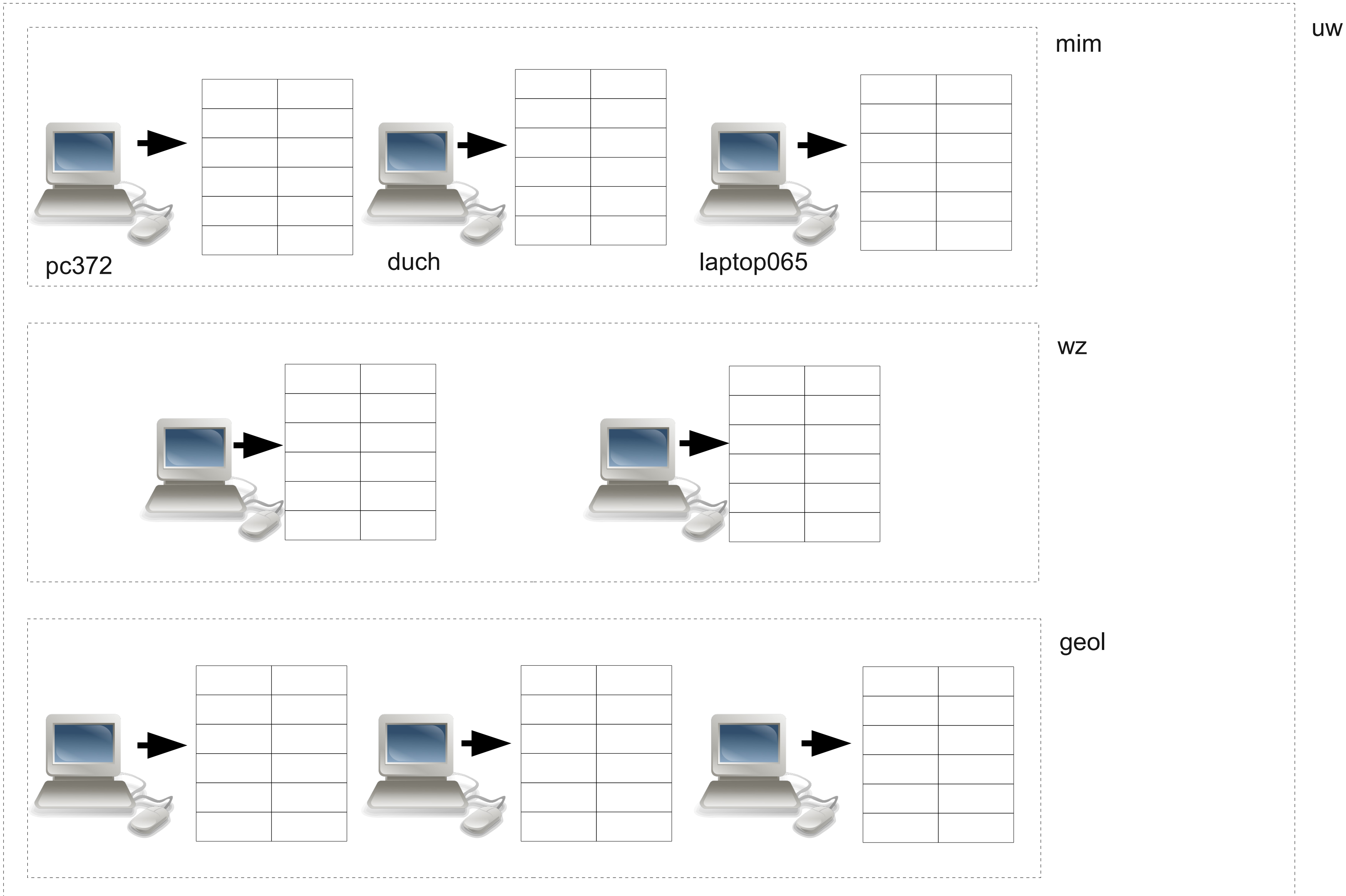
Zone Hierarchy

Zone Hierarchy

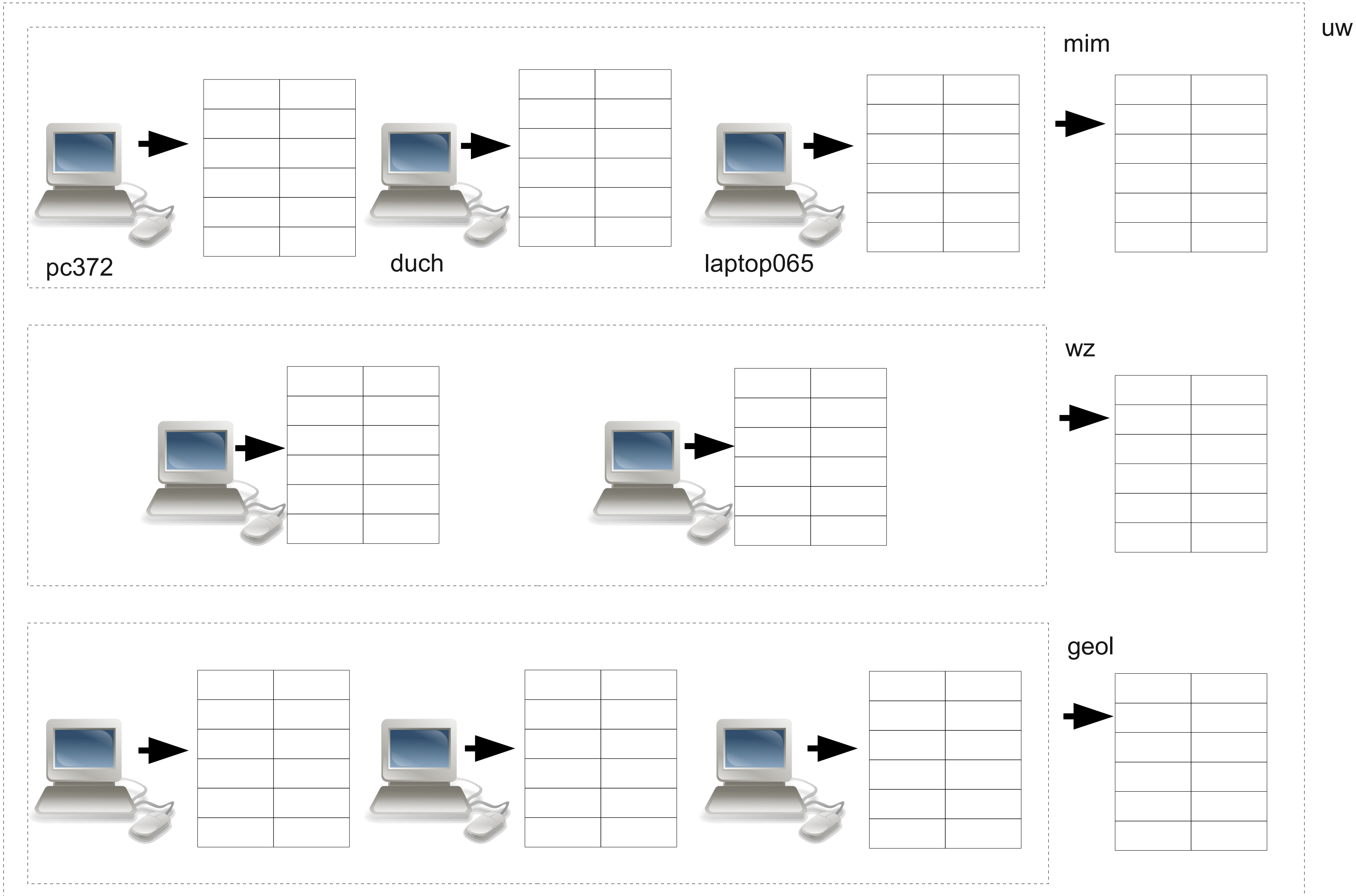


Zone Hierarchy

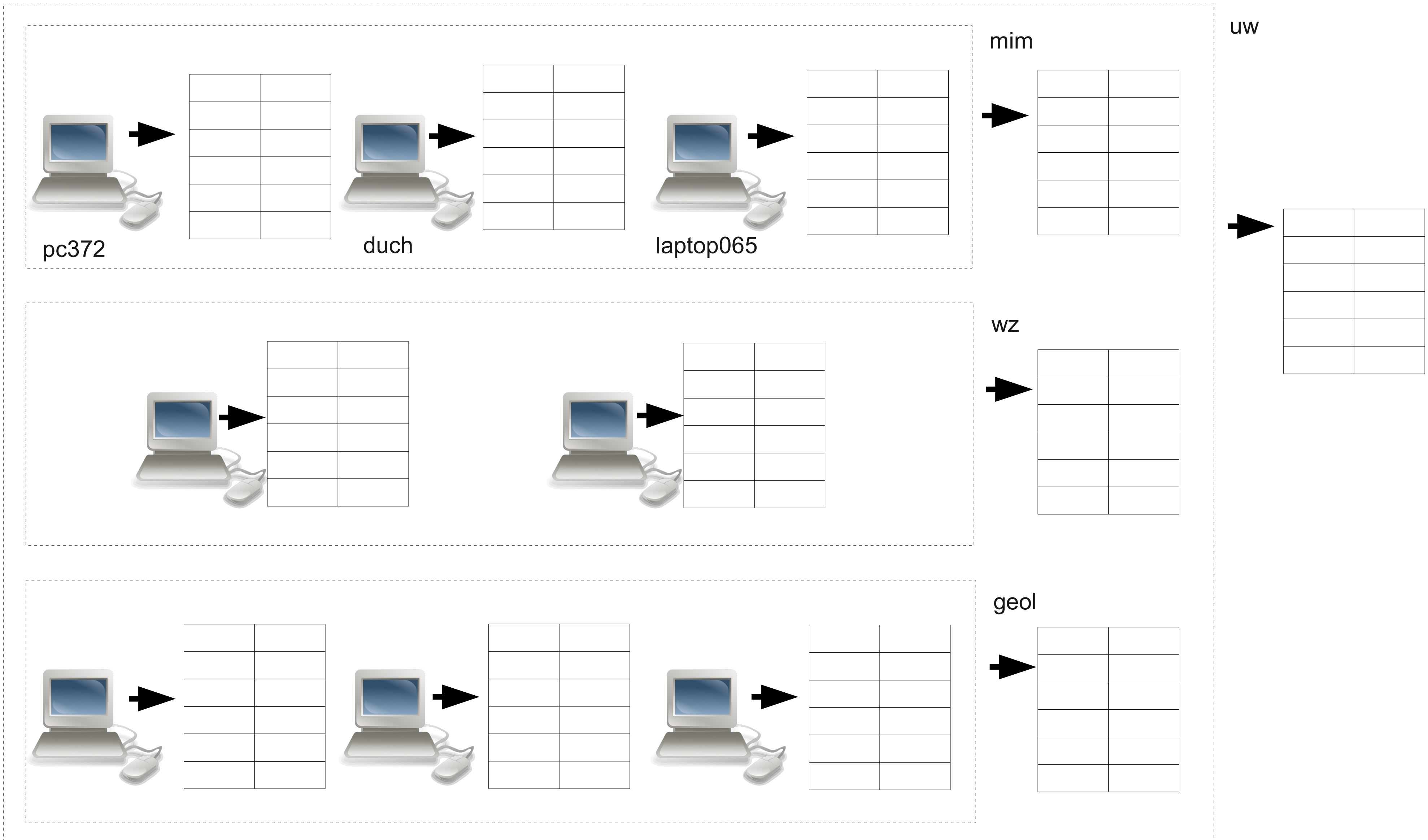
Attribute List



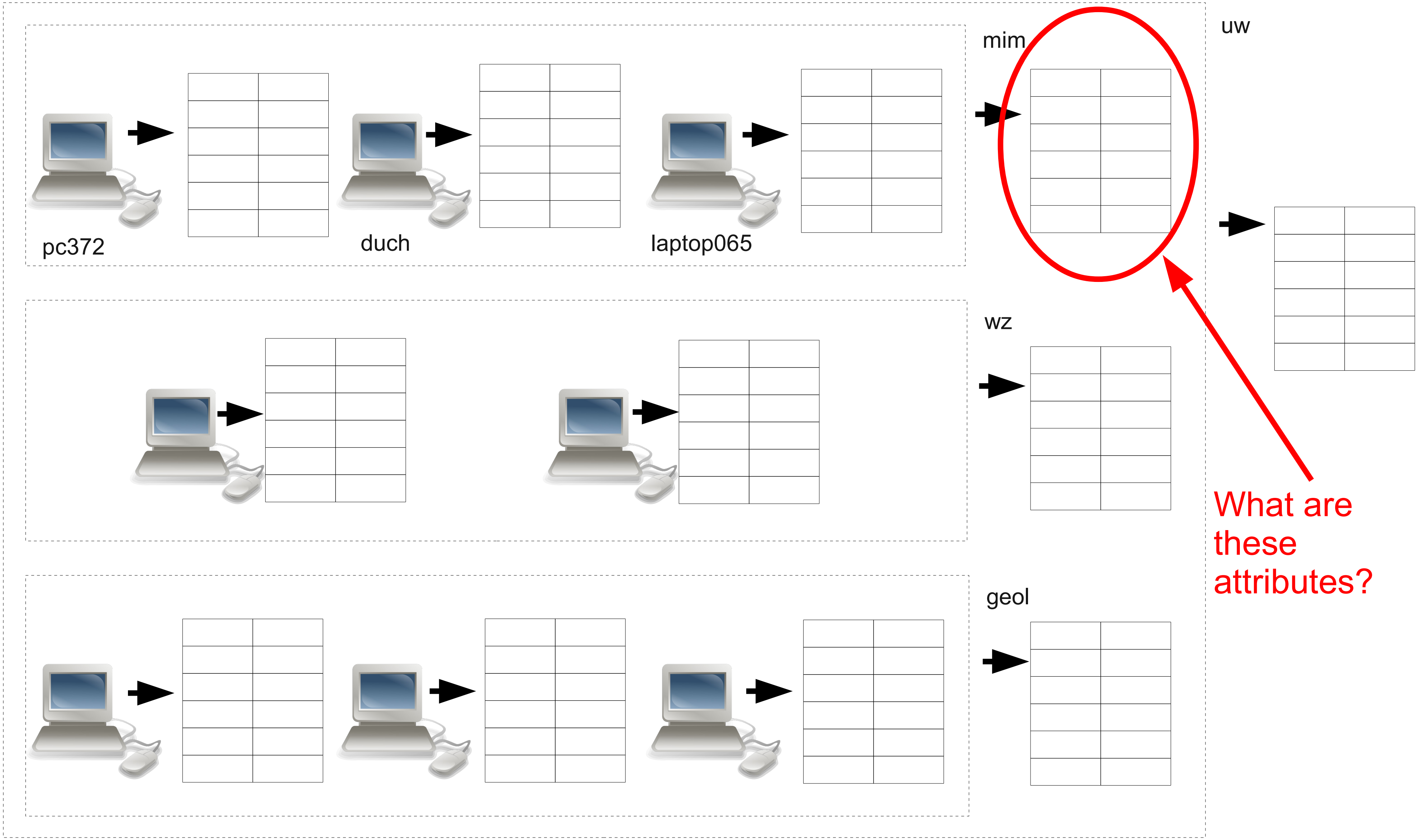
Attribute List



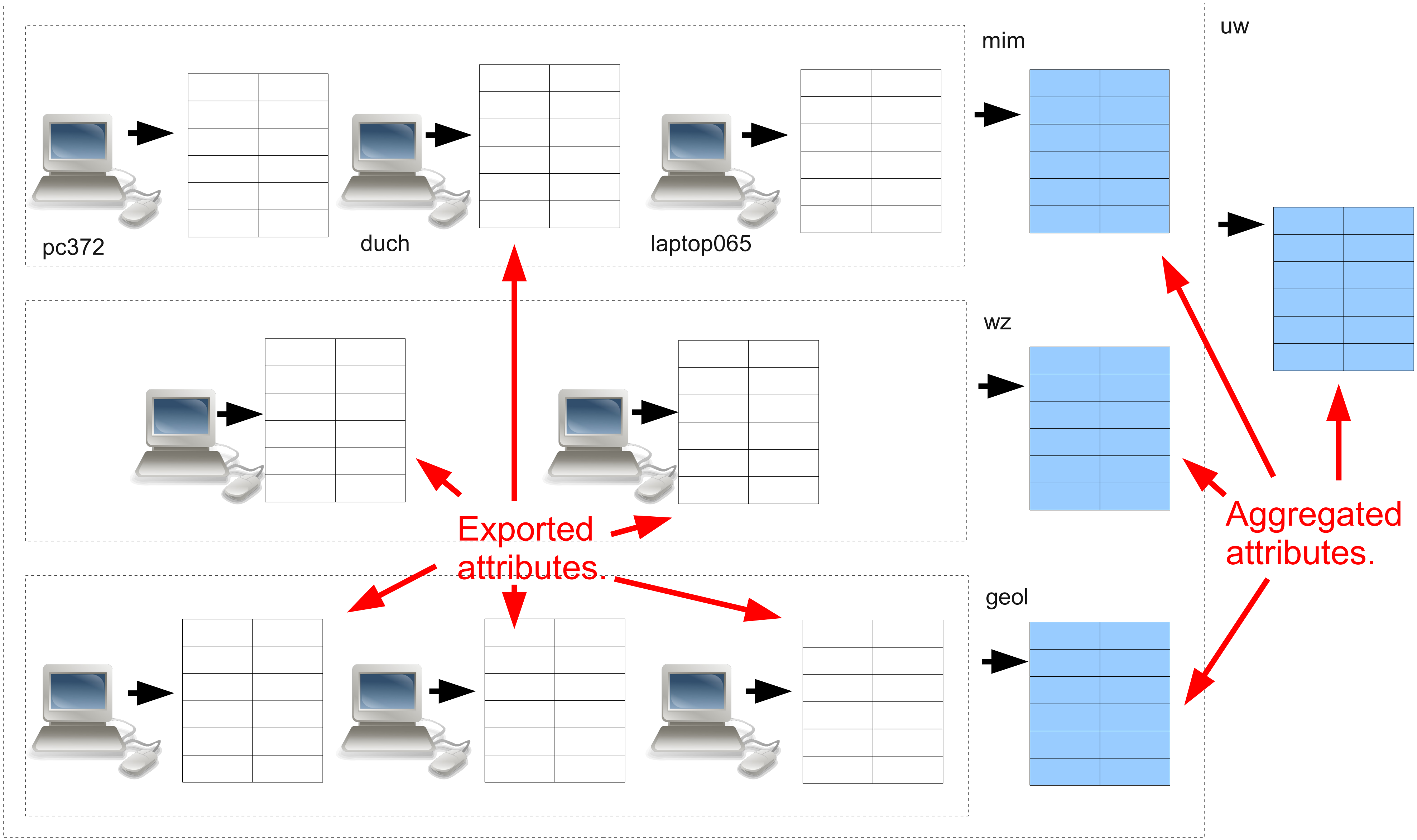
Attribute List



Attribute List



Aggregation Functions



Aggregation Functions

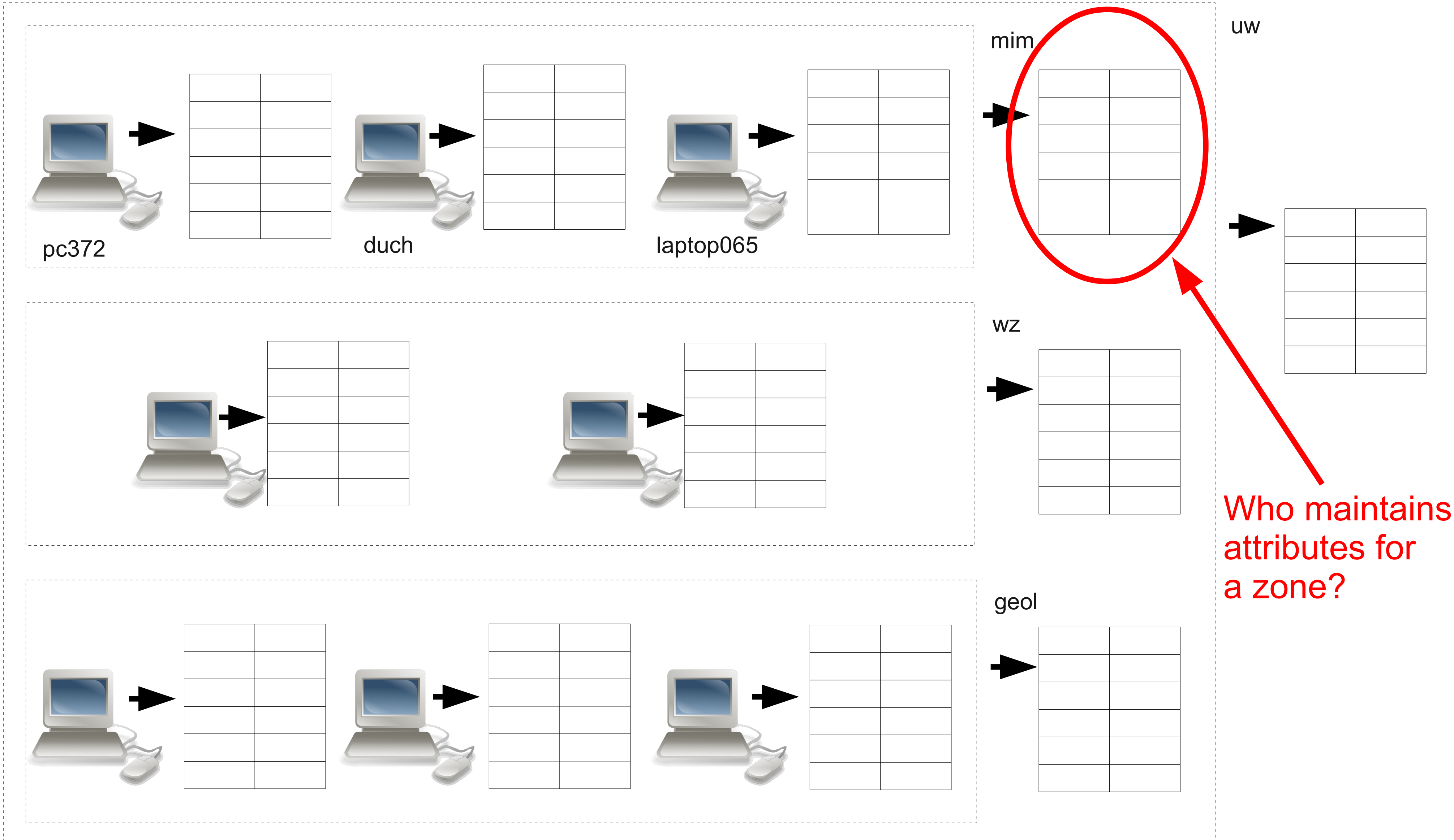
Aggregated attributes a continuously recomputed.

Function	Description
MIN(attribute)	Find the minimum attribute.
MIN(attribute)	Find the maximum attribute.
SUM(attribute)	Sum the attributes.
COUNT(attribute)	Compute the attributes.
AVG(attribute [, weight])	Calculate a weighted average.
OR(attribute)	Bit-wise OR of a bitmap.
AND(attribute)	Bit-wise AND of a bitmap.
FIRST(n, attribute)	Return a set with the first n attributes.
RANDOM(n, attribute [, weight])	Return a set with n randomly selected attributes.

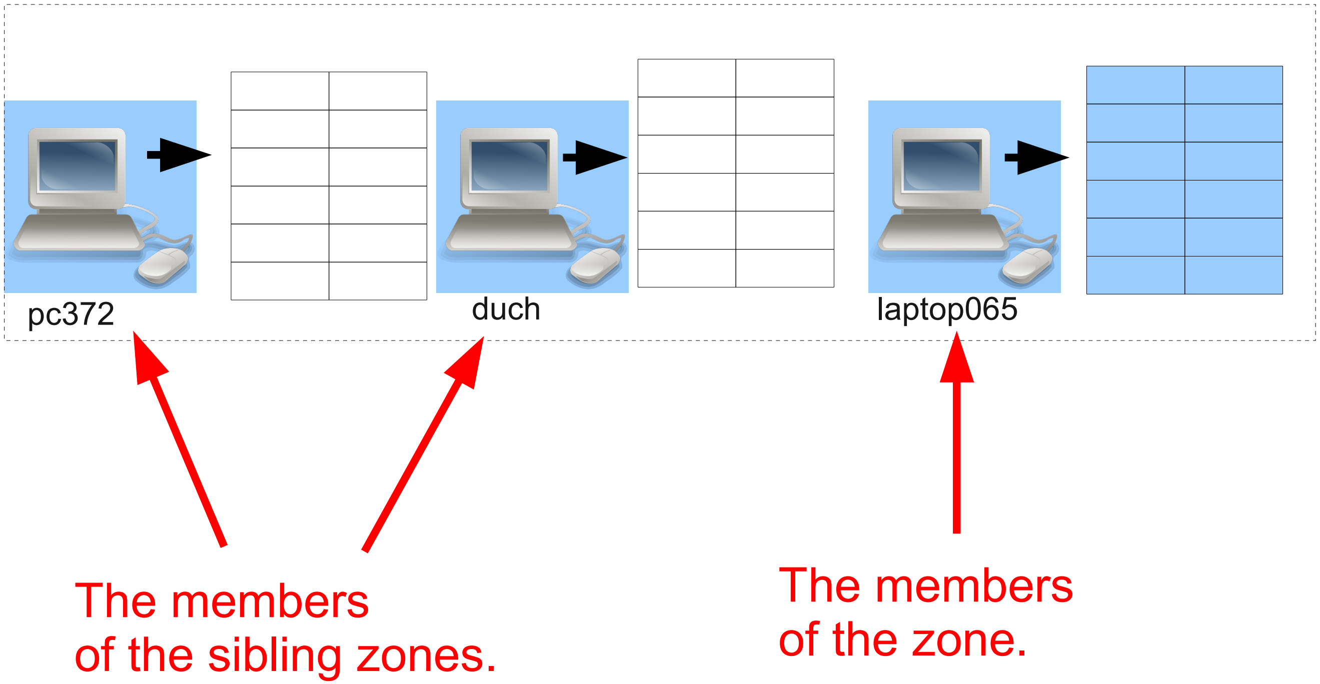
In general, aggregation functions should compact N values into a small output value.

The size of the output value should be a small function of N:
- preferably O(1),
- alternatively O(logN).

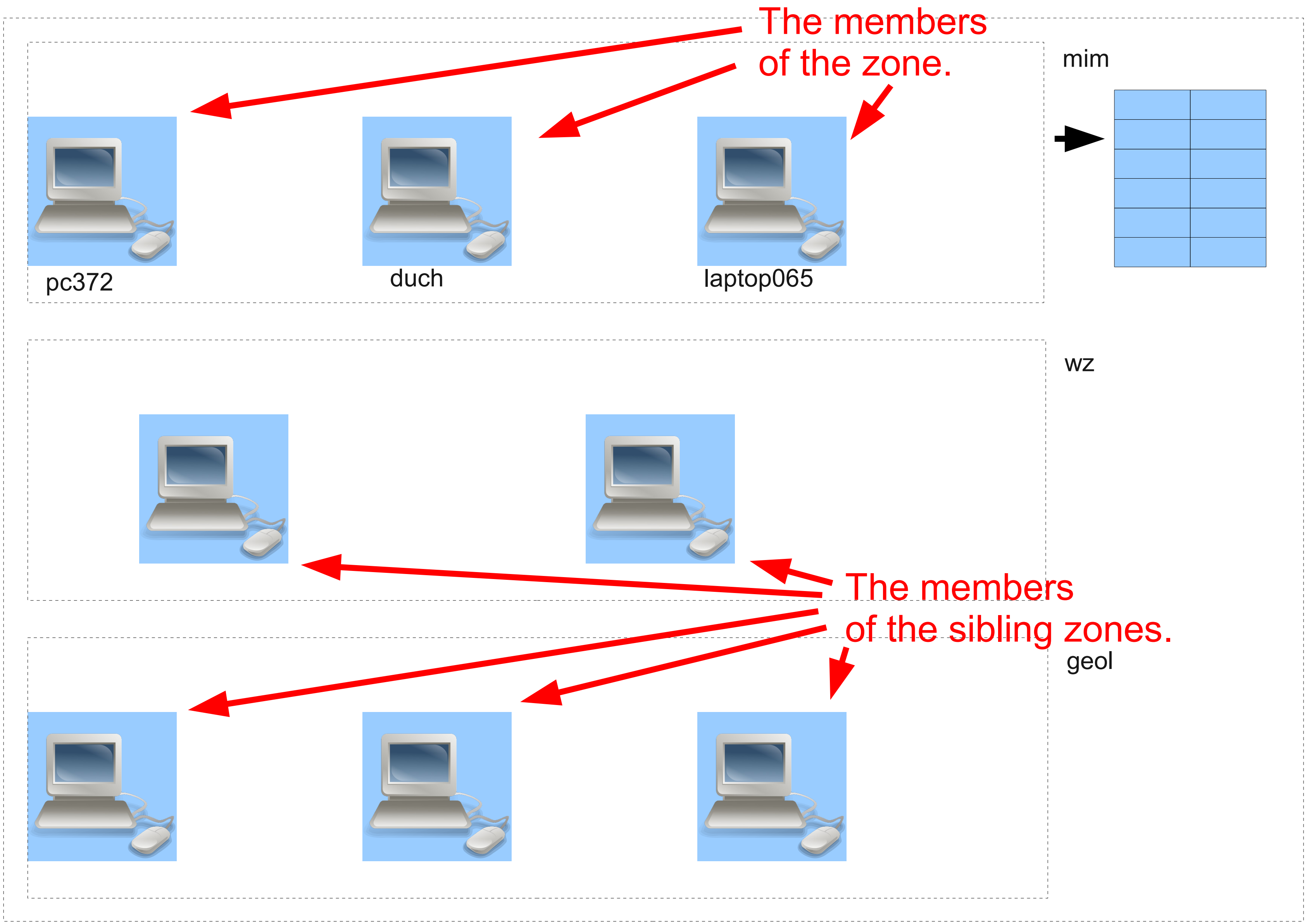
Attribute Maintenance



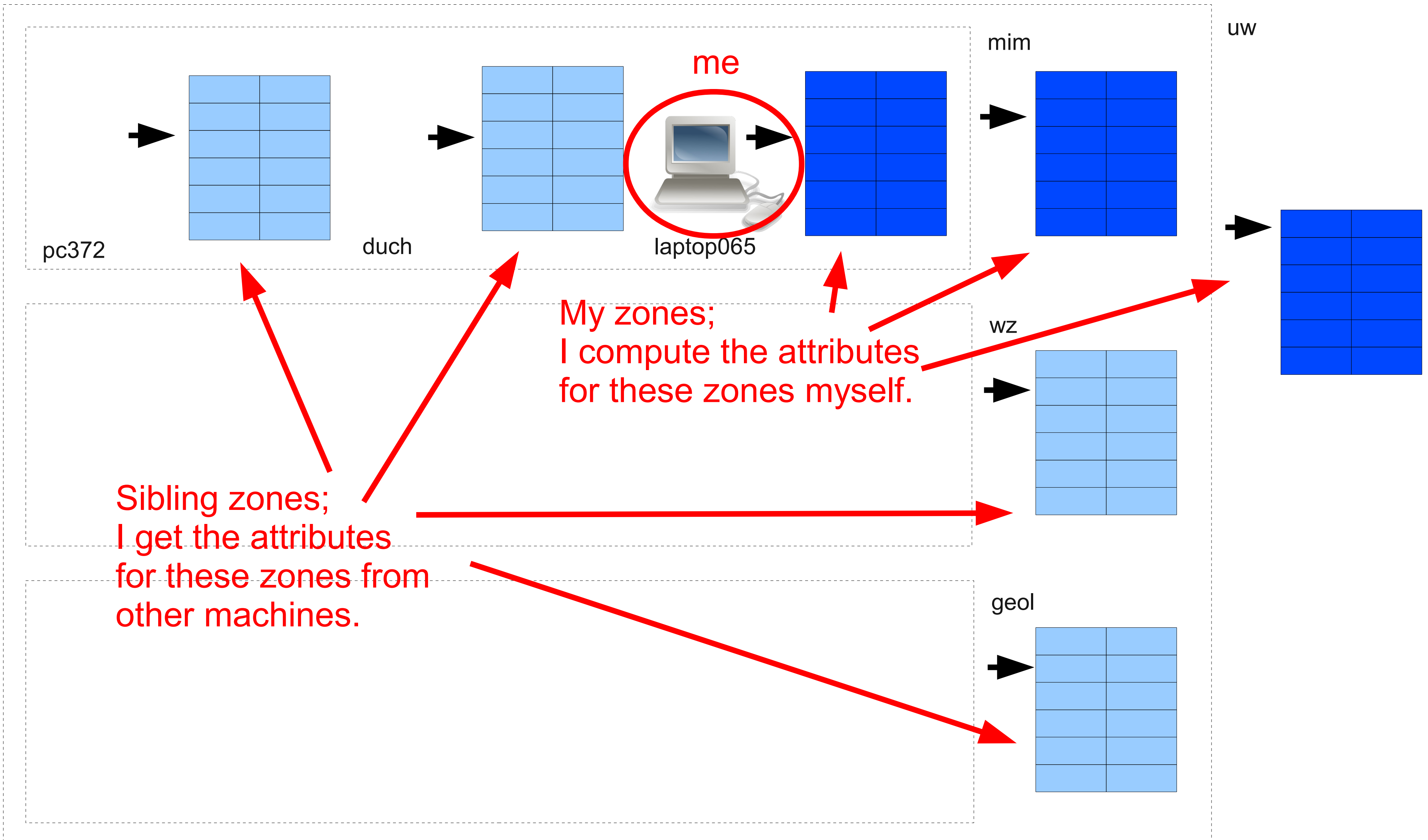
Attribute Maintenance



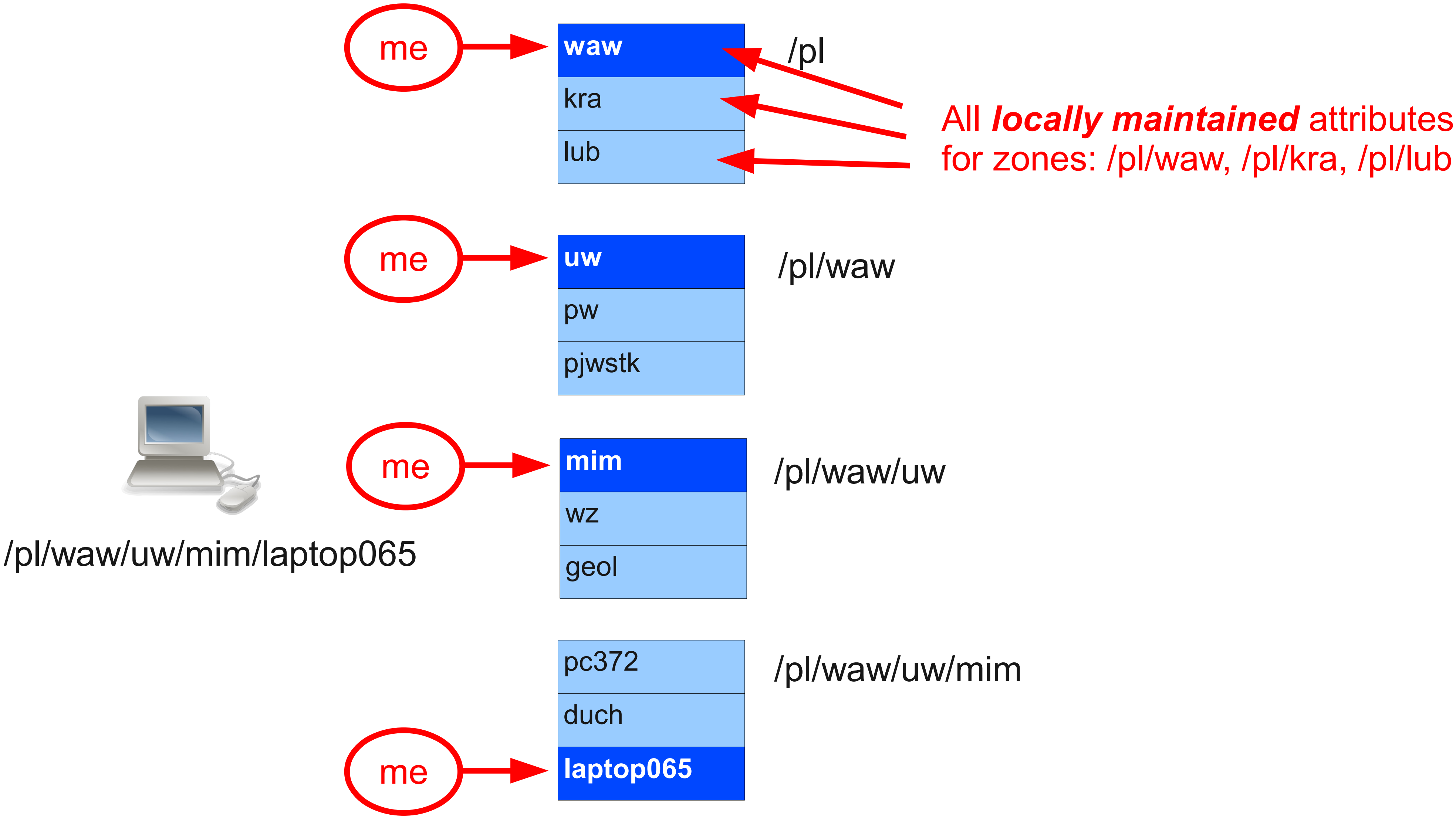
Attribute Maintenance



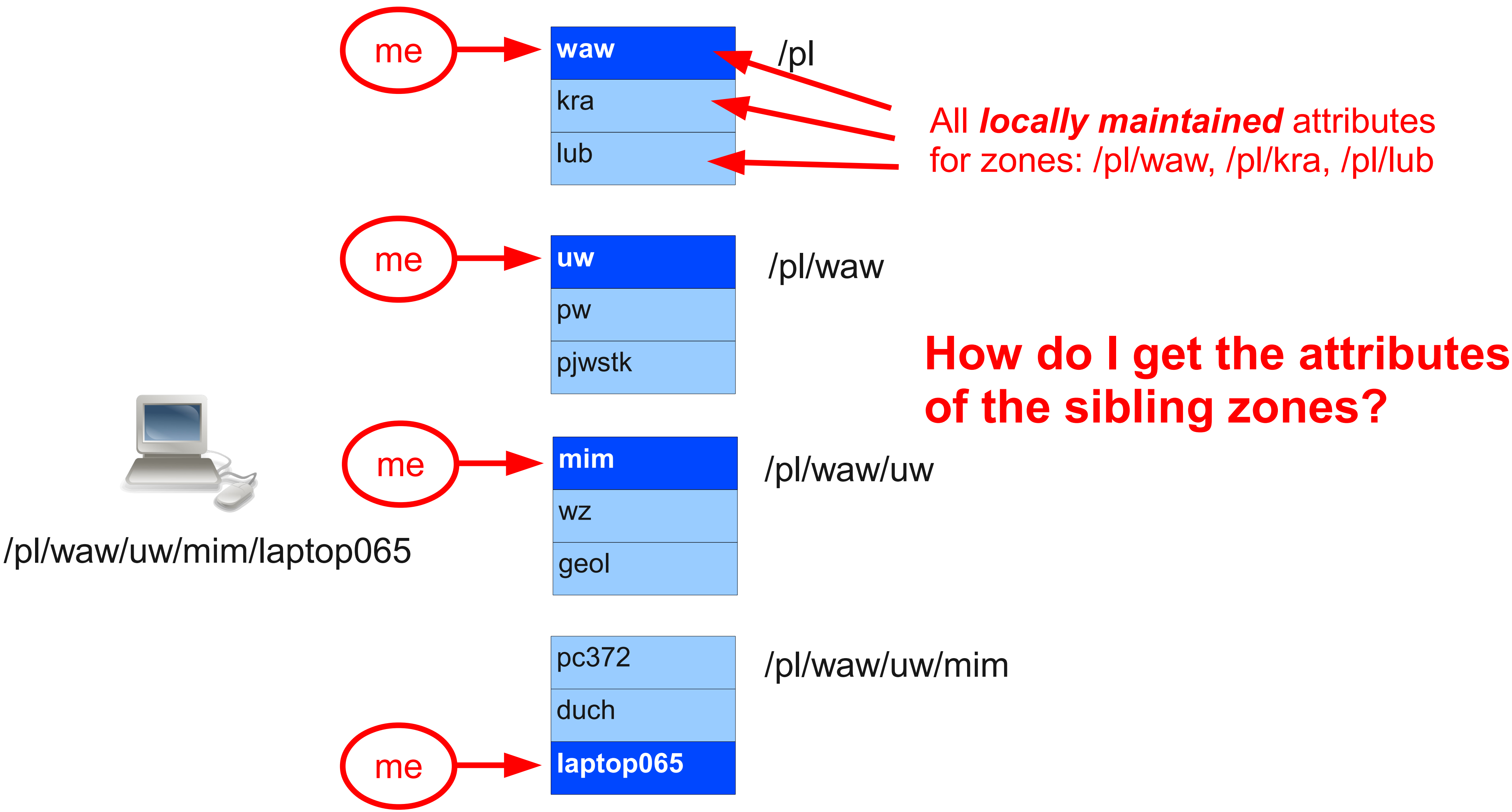
Attribute Maintenance



Attribute Maintenance



Attribute Maintenance



Hierarchical Gossiping

waw	null
kra	147.13.132.137, ...
lub	182..132.137, ...

uw	null
pw	195.28.56.201, ...
pjwstk	162.32.90.45, ...

mim	null
wz	211.123.1.15, ...
geol	193.14.16.29, ...

pc372	193.0.96.127
duch	193.0.96.2
laptop065	193.0.96.415

/pl/waw/uw/mim/laptop065

One of the attributes maintained Locally for a zone is ***a list of contacts for the zone***.

A list of contacts for zone /pl/kra

A list of contacts for zone /pl/waw/uw/geol

Hierarchical Gossiping

- A computer in zone */Z/X computes the attributes for */Z/X locally, based on the child zone attributes.
- To obtain attributes for a sibling zone, */Z/Y, the computer:
  - Selects a random contact IP with zone */Z/Y.
  - Communicates with this IP to exchange:
    - The attributes of all child zones of the parent zone, */Z.
    - The attributes of all common higher level zones.
  - Suppose that A_{local} is the local (the computer's) value of attribute A, while A_{remote} is the remote (the contact's) value of A, as obtained in the exchange. The computer chooses the **fresher** attribute:
    - If $TimeComputed(A_{local}) < TimeComputed(A_{remote})$: $A_{local} := A_{remote}$

Hierarchical Gossiping

waw	null
kra	147.13.132.137, ...
lub	182..132.137, ...

uw	null
pw	195.28.56.201, ...
pjwstk	162.32.90.45, ...

mim	null
wz	211.123.1.15, ...
geol	193.14.16.29, ...

pc372	193.0.96.127
duch	193.0.96.2
laptop065	193.0.96.415

- Select zone to gossip for.

/pl/waw/uw/wz

/pl/waw/uw/mim/laptop065

Hierarchical Gossiping

waw	null
kra	147.13.132.137, ...
lub	182..132.137, ...

uw	null
pw	195.28.56.201, ...
pjwstk	162.32.90.45, ...

mim	null
wz	**211.123.1.15**, ...
geol	193.14.16.29, ...

pc372	193.0.96.127
duch	193.0.96.2
laptop065	193.0.96.415

- Select zone to gossip for.
- Randomly select a contact IP.

/pl/waw/uw/mim/laptop065

Hierarchical Gossiping

waw	null
kra	147.13.132.137, ...
lub	182..132.137, ...

uw	null
pw	195.28.56.201, ...
pjwstk	162.32.90.45, ...

mim	null
wz	211.123.1.15, ...
geol	193.14.16.29, ...

pc372	193.0.96.127
duch	193.0.96.2
laptop065	193.0.96.415

- Select zone to gossip for.
- Randomly select a contact IP.
- Exchange common zone attributes.

Exchange the common attributes with 211.123.1.15

/pl/waw/uw/mim/laptop065

Hierarchical Gossiping

waw	null
kra	147.13.132.137, ...
lub	182..132.137, ...

uw	null
pw	195.28.56.201, ...
pjwstk	162.32.90.45, ...

mim	null
wz	211.123.1.15, ...
geol	193.14.16.29, ...

pc372	193.0.96.127
duch	193.0.96.2
laptop065	193.0.96.415

- Select zone to gossip for.
- Randomly select a contact IP.
- Exchange common zone attributes.
- Adopt the fresher attributes.

/pl/waw/uw/mim/laptop065

Hierarchical Gossiping

waw	null
kra	147.13.132.137, ...
lub	182..132.137, ...

uw	null
pw	195.28.56.201, ...
pjwstk	162.32.90.45, ...

mim	null
wz	211.123.1.15, ...
geol	193.14.16.29, ...

pc372	193.0.96.127
duch	193.0.96.2
laptop065	193.0.96.415

/pl/waw/uw/mim/laptop065

- Select zone to gossip for.
- Randomly select a contact IP.
- Exchange common zone attributes.
- Adopt the fresher attributes.
- Mine and the contact's exchanged attributes are now the same.

Hierarchical Gossiping

- The gossiping process is continuous.
  - Each computer initiates a gossip every 5 seconds.
  - It also receives gossip requests from other computers.
  - The gossip is performed at every level of the zone hierarchy.

Hierarchical Gossiping

- The gossiping process is continuous.
  - Each computer initiates a gossip every 5 seconds.
  - It also receives gossip requests from other computers.
  - The gossip is performed at every level of the zone hierarchy.
- If the attribute values did not change, the computers would all reach a consistent view:
  - ***Eventual consistency.***.

Hierarchical Gossiping

- The gossiping process is continuous.
  - Each computer initiates a gossip every 5 seconds.
  - It also receives gossip requests from other computers.
  - The gossip is performed at every level of the zone hierarchy.
- If the attribute values did not change, the computers would all reach a consistent view:
  - ***Eventual consistency.***
- Gossiping is extremely ***robust*** to node failures and changes in connectivity:
  - If a computer to contact with dies, simply another contact can be chosen at random.

Hierarchical Gossiping

- How do computers know zone contact IPs?

Hierarchical Gossiping

- How do computers know zone contact IPs?
- They are simply gossiped like other attributes (details in the paper).

Usability

- The aggregation functions and the attributes are not static.
  - They can be dynamically installed in the zones for which we are interested in some attribute values.
  - They propagate automatically to the sibling and parent zones (via gossiping).

Usability

- The aggregation functions and the attributes are not static.
  - They can be dynamically installed in the zones for which we are interested in some attribute values.
  - They propagate automatically to the sibling and parent zones (via gossiping).
- This is a form of ***mobile code***.

Usability

- The aggregation functions and the attributes are not static.
  - They can be dynamically installed in the zones for which we are interested in some attribute values.
  - They propagate automatically to the sibling and parent zones (via gossiping).
- This is a form of ***mobile code***.
- In this way, not only can we monitor aggregate information about resources, but we can also locate particular resources...
  - … and do other interesting stuff (details in the paper).

Security

- Since aggregation functions are dynamically installed, we have to ensure some security.

Security

- Since aggregation functions are dynamically installed, we have to ensure some security.
- The aggregation functions require certificates issued by a trusted certification authority.

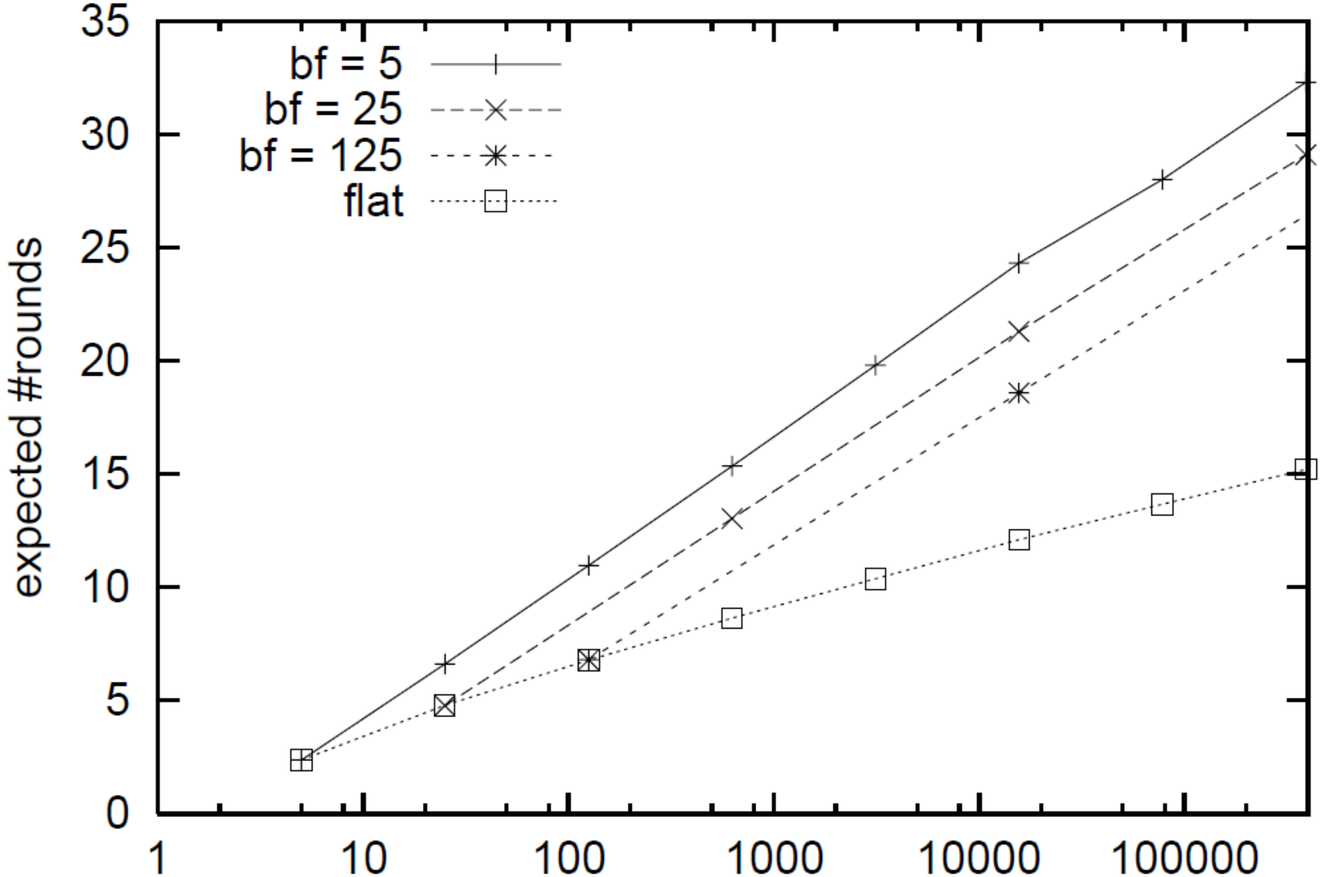
Security

- Since aggregation functions are dynamically installed, we have to ensure some security.
- The aggregation functions require certificates issued by a trusted certification authority.
- An aggregation function or an attribute can be installed in a zone only if it is accompanied by a certificate for that zone.

Security

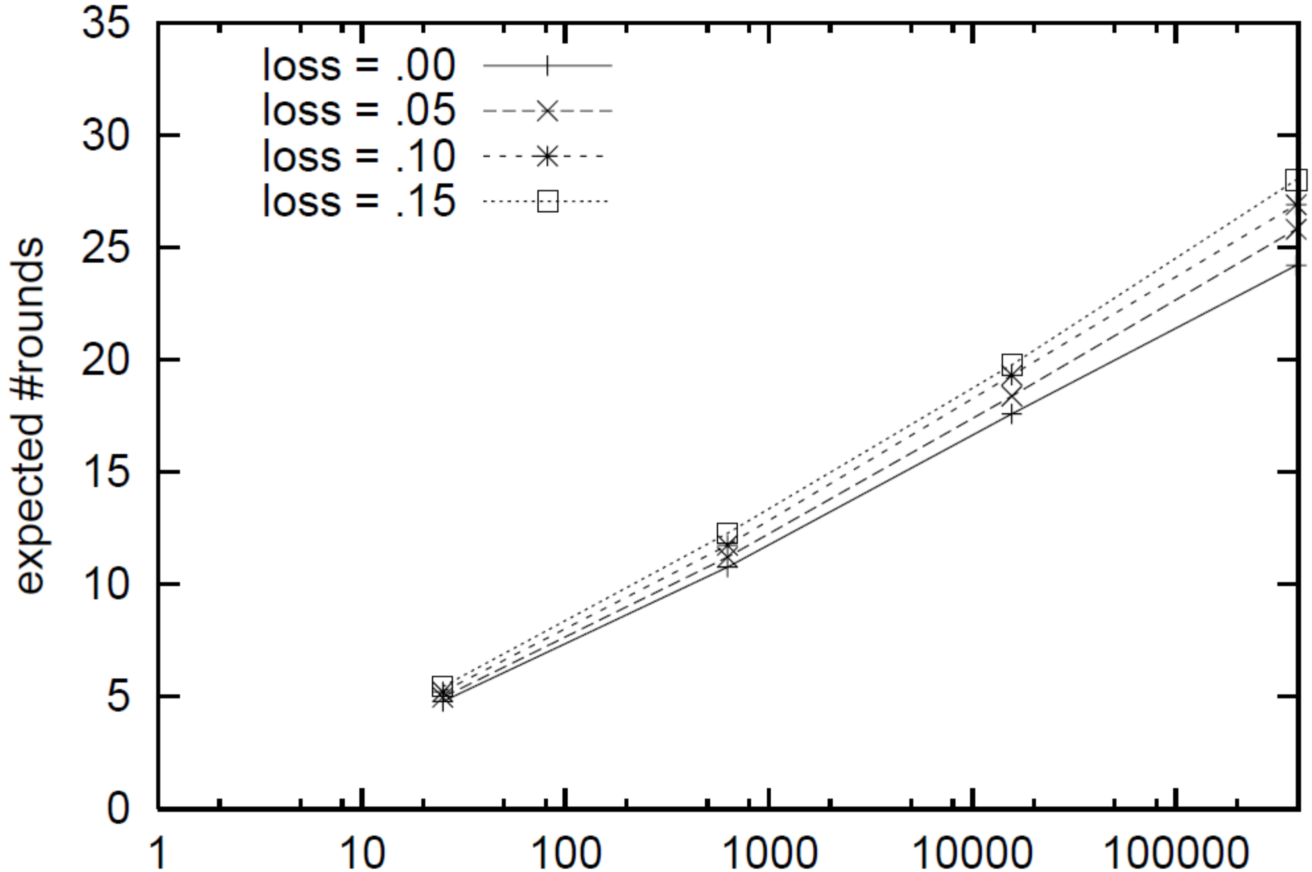
- Since aggregation functions are dynamically installed, we have to ensure some security.
- The aggregation functions require certificates issued by a trusted certification authority.
- An aggregation function or an attribute can be installed in a zone only if it is accompanied by a certificate for that zone.
- In this way, Astrolabe ensures write access control and integrity (details in the paper).

Performance



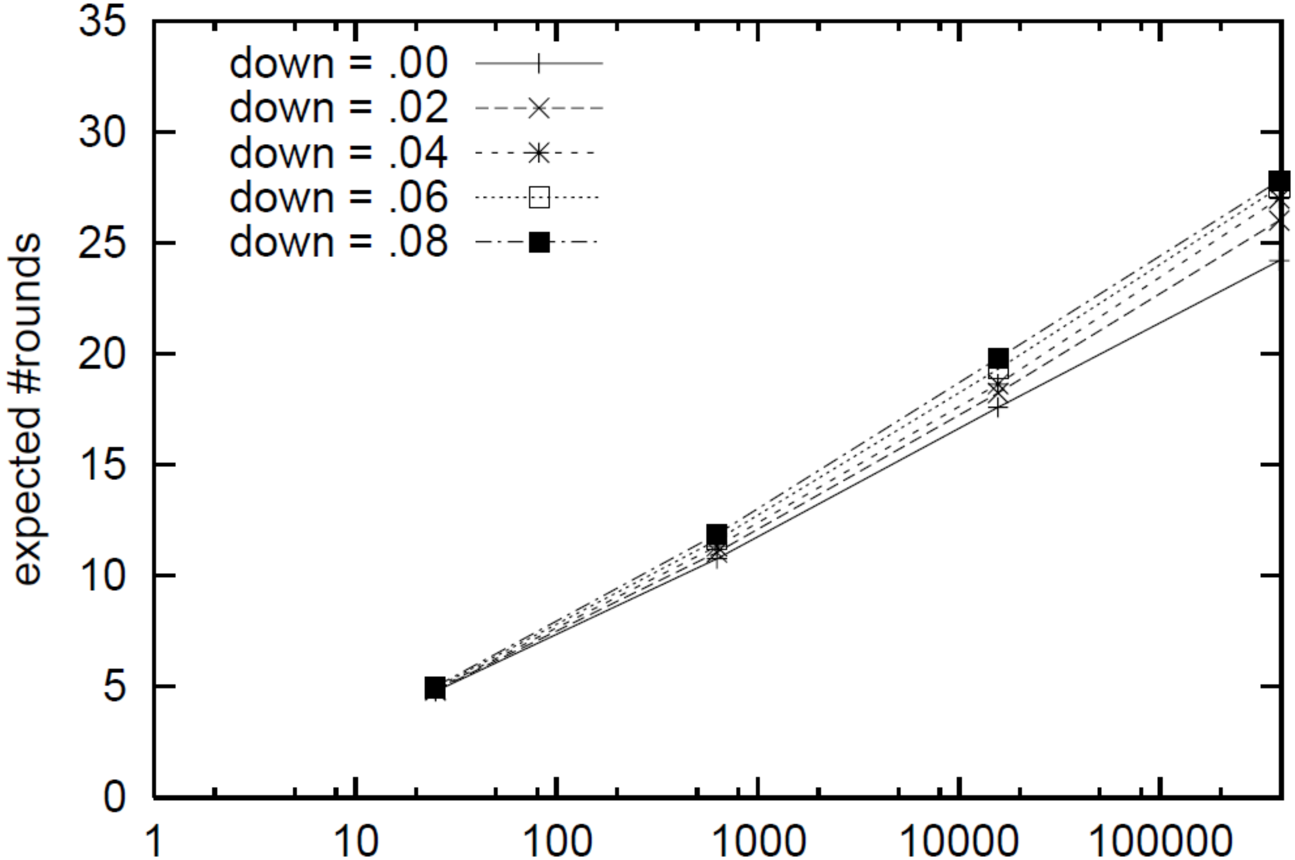
The average number of rounds to propagate an update depending on the number of sibling zones at every hierarchy level.

Performance



The average number of rounds to propagate an update depending on the probability of a message being lost (bf = 25).

Performance



The average number of rounds to propagate an update depending on the probability of a node being down (bf = 25).

Summary

- The goal of Astrolabe is to manage a large collection of distributed objects and resources.
- Astrolabe's design employs the following principles:
  - **Scalability through hierarchy**:
    - Zone hierarchy, hierarchical attribute aggregation
  - **Flexibility through mobile code**:
    - Dynamically installed aggregation functions
  - **Robustness via a gossip-based peer-to-peer protocol**:
    - Self-management and recovery
  - **Security through certificates**:
    - Integrity and write access control

Thank You

Questions?