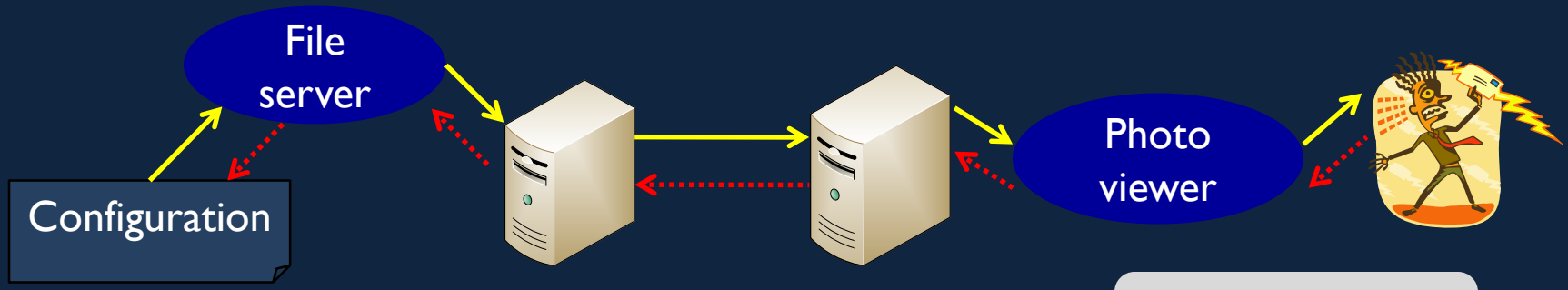
Detailed and understandable network diagnosis

Ratul Mahajan

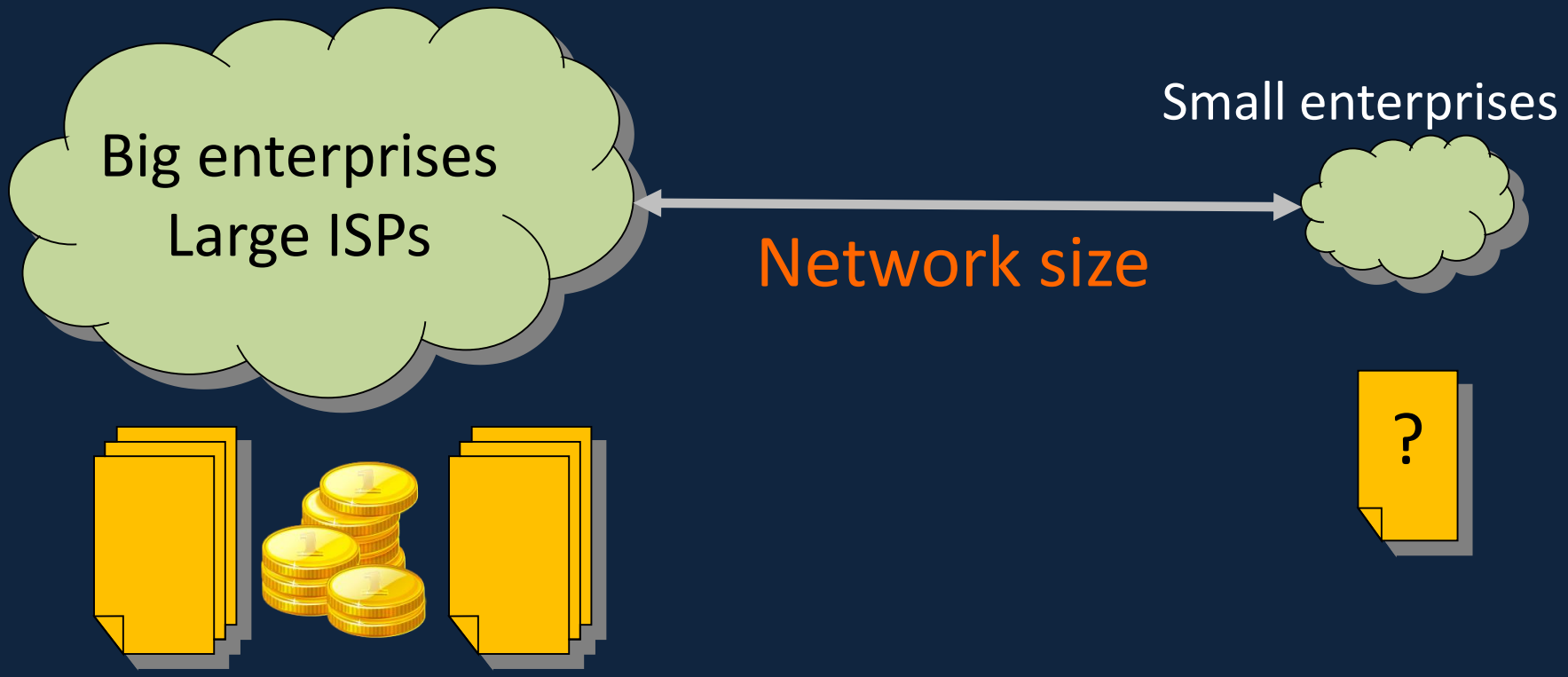
With Srikanth Kandula, Bongshin Lee,
Zhicheng Liu *(GaTech),* Patrick Verkaik *(UCSD)*,
Sharad Agarwal, Jitu Padhye, Victor Bahl

Network diagnosis explains faulty behavior

Starts with problem symptoms and ends at likely culprits



Current landscape of network diagnosis systems

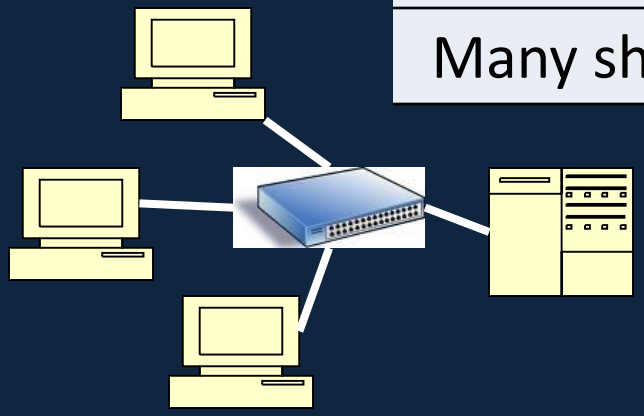


Why study small enterprise networks separately?

Big enterprises
Large ISPs

Small enterprises

Less sophisticated admins
Less rich connectivity
Many shared components



IIS, SQL,
Exchange, …

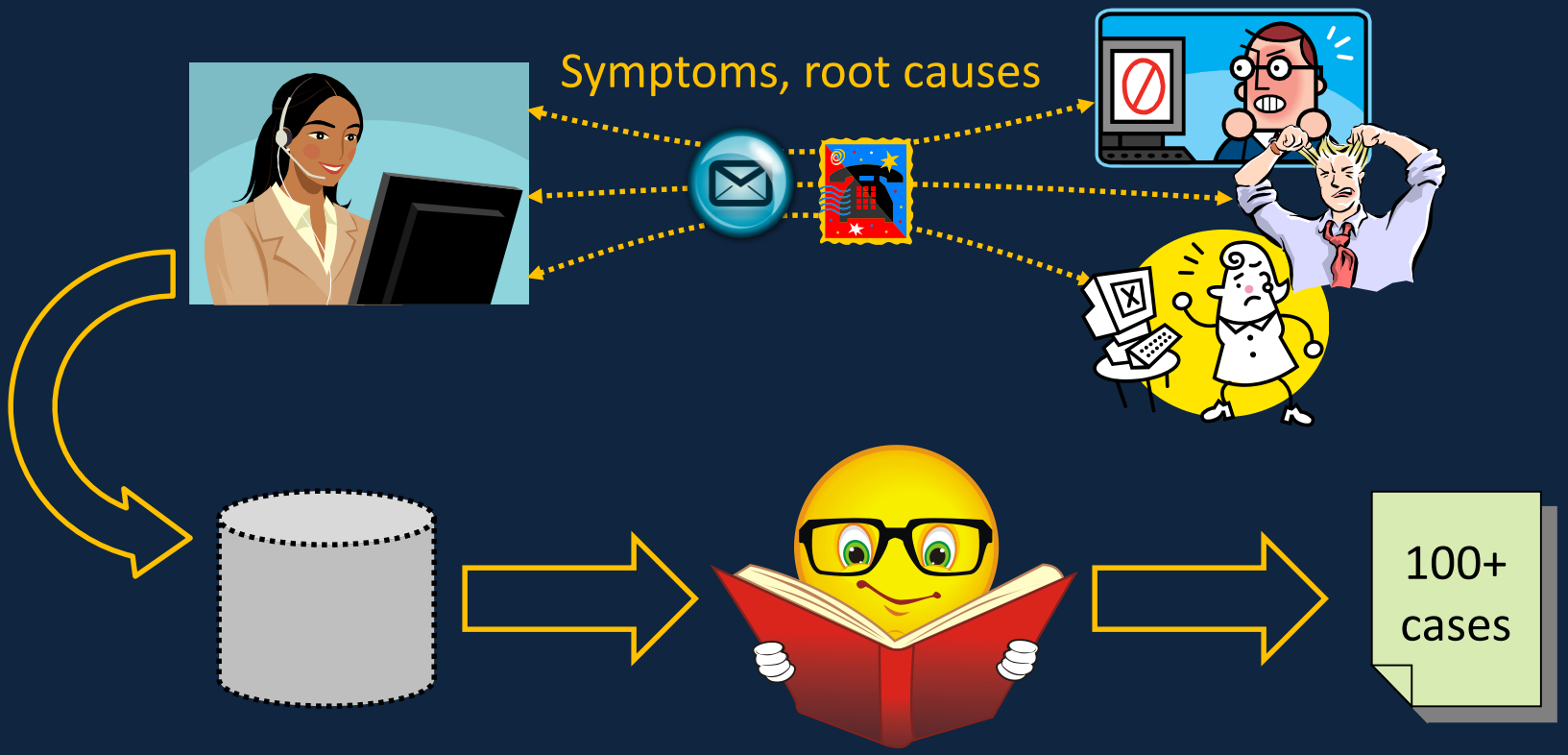
Our work

1. Uncovers the need for detailed and understandable diagnosis

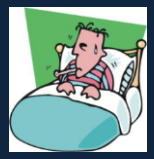
2. Develops *NetMedic* for detailed diagnosis
   - Diagnoses application faults without application knowledge

3. Develops *NetClinic* for explaining diagnostic analysis

Understanding problems in small enterprises

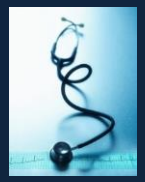


And the survey says .....



Symptom

App-specific	60 %
Failed initialization	13 %
Poor performance	10 %
Hang or crash	10 %
Unreachability	7 %



Identified cause

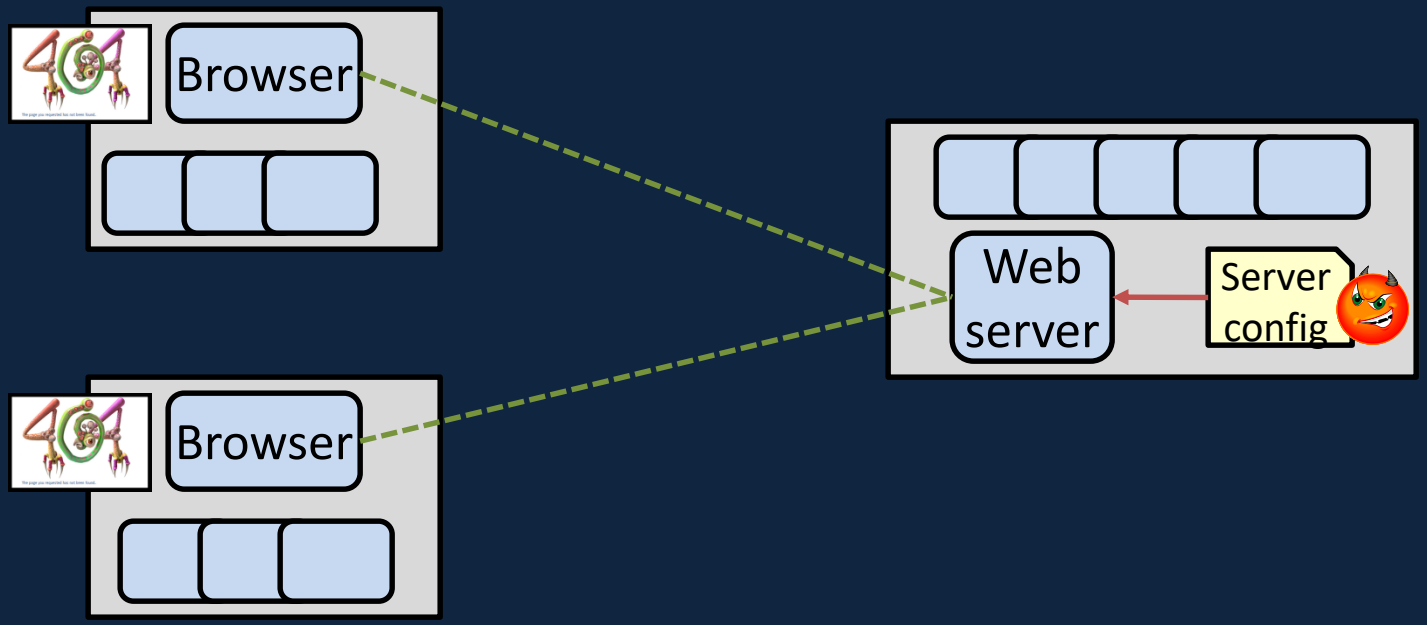
Non-app config (e.g., firewall)	30 %
Software/driver bug	21 %
App config	19 %
Overload	4 %
Hardware fault	2 %

Handle app-specific as well as generic faults

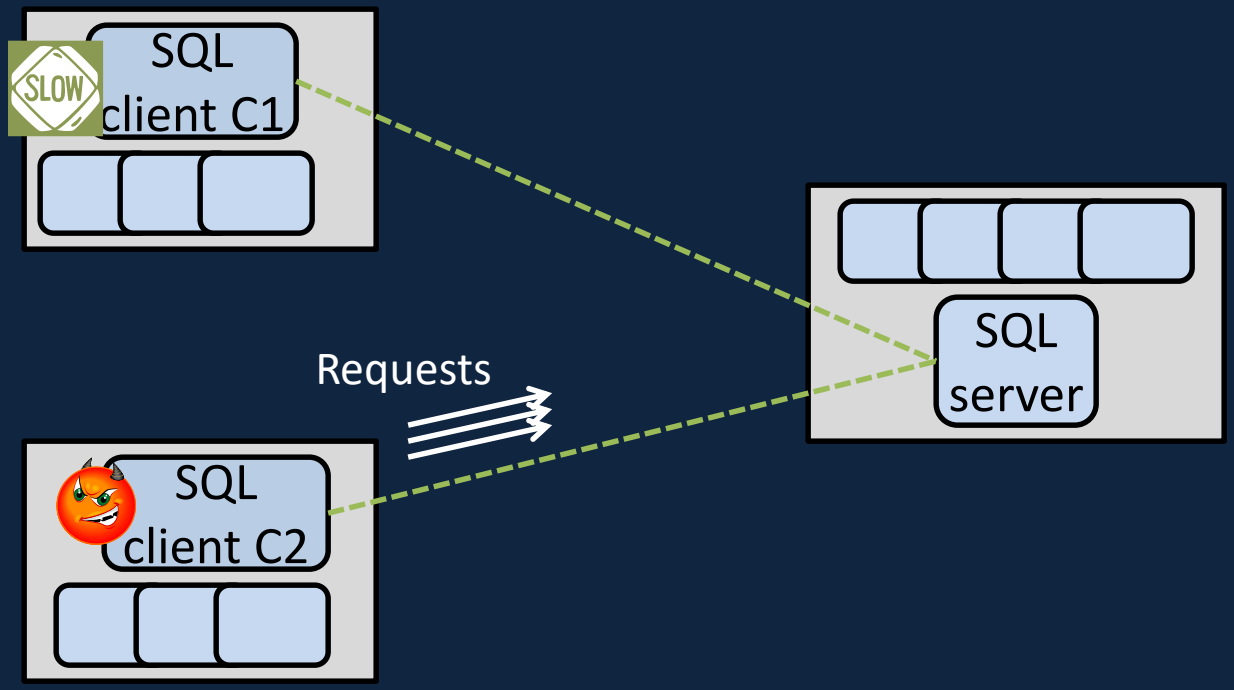
Identify culprits at a fine granularity

Detailed diagnosis

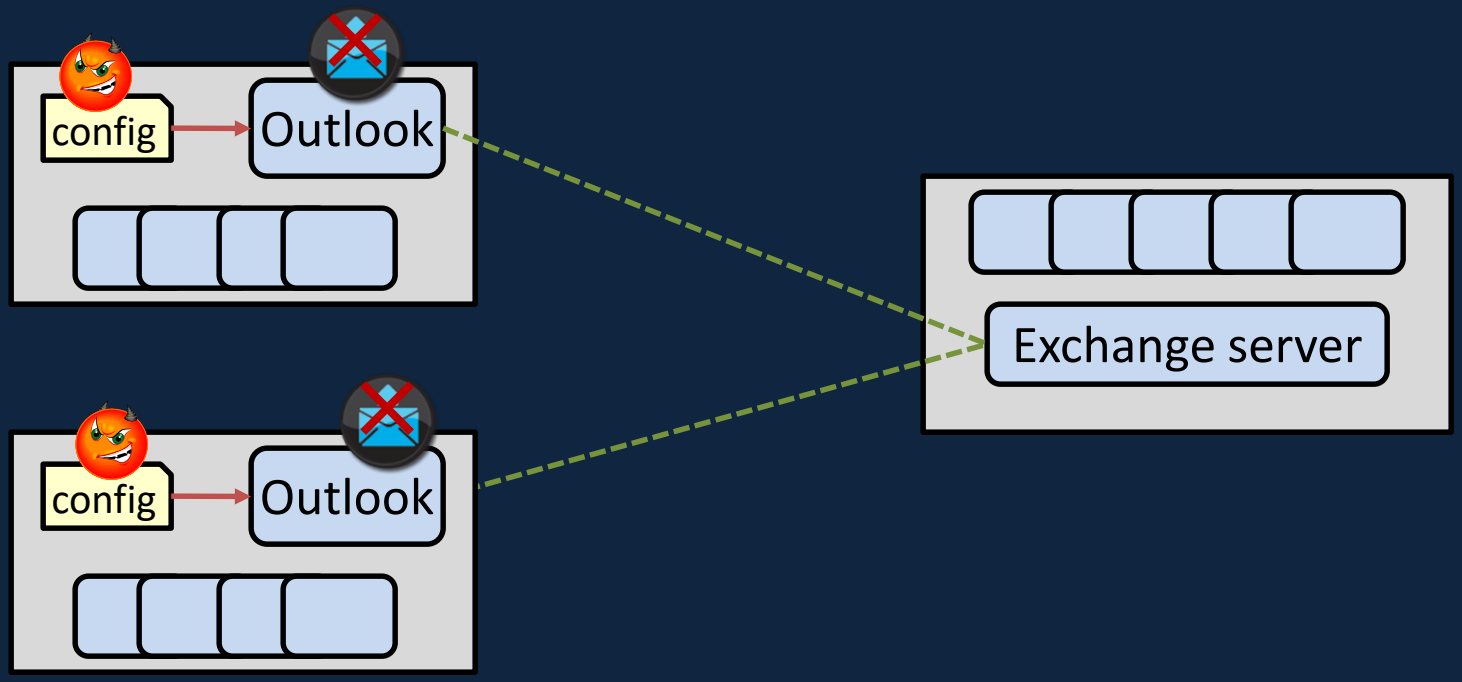
Example problem 1: Server misconfig



Example problem 2: Buggy client



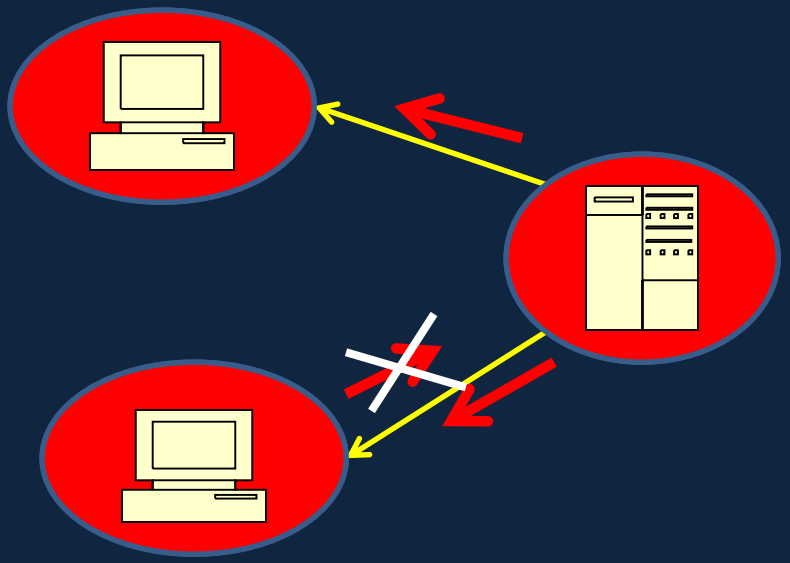
Example problem 3: Client misconfig



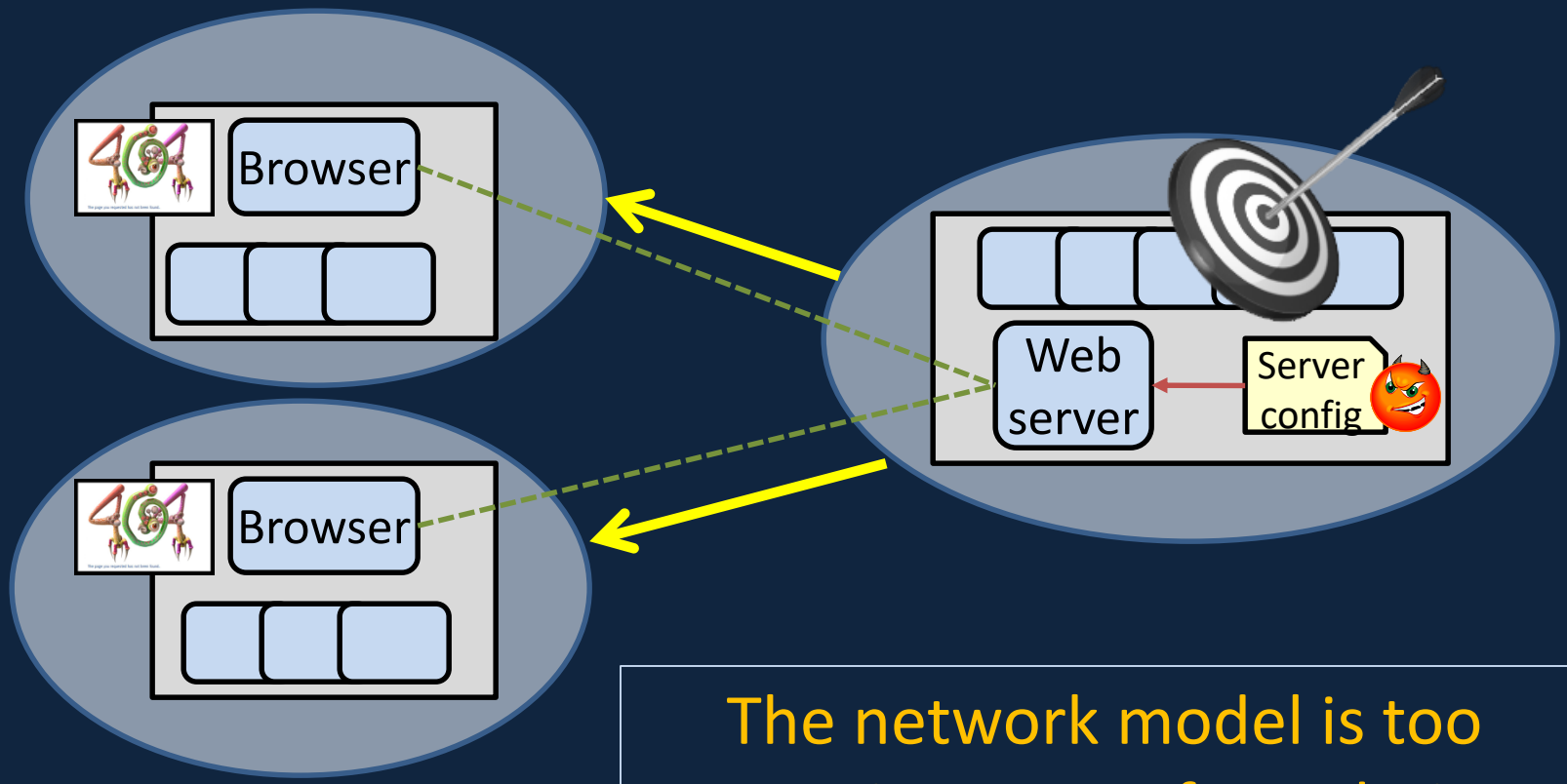
Current formulations sacrifice detail (to scale)

Dependency graph based formulations (e.g., Sherlock [SIGCOMM2007])

- Model the network as a dependency graph at a coarse level
- Simple dependency model

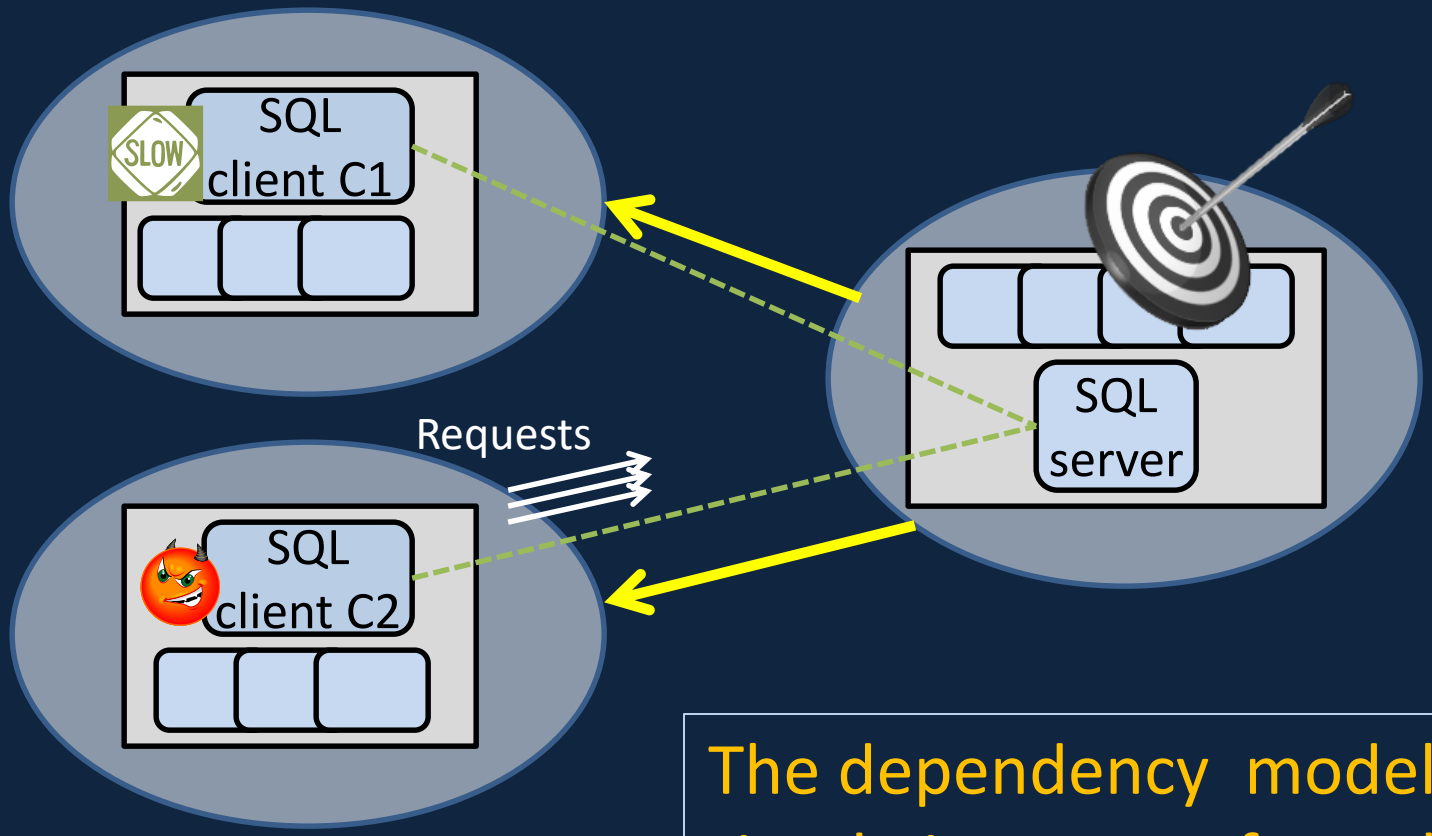


Example problem 1: Server misconfig



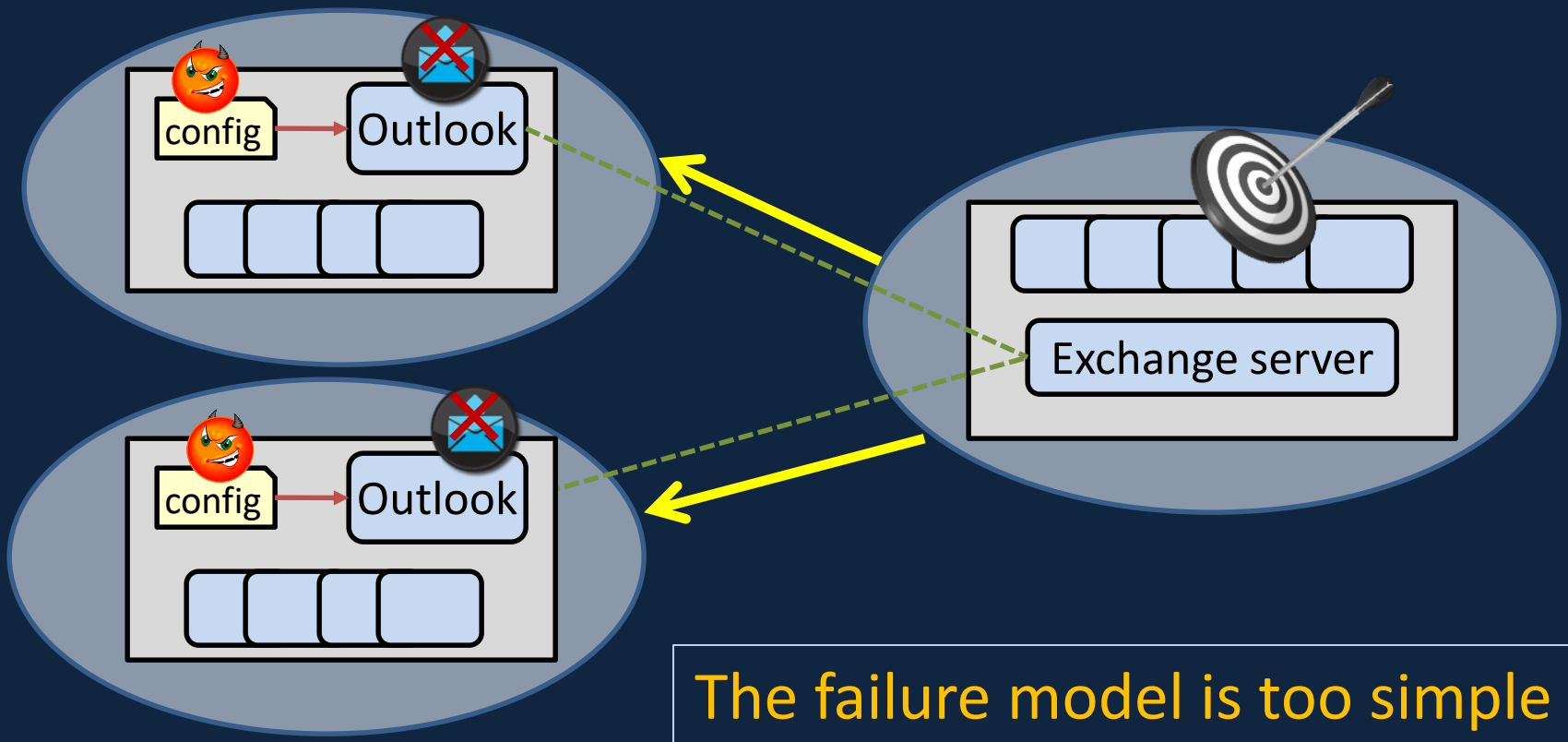
The network model is too coarse in current formulations

Example problem 2: Buggy client



The dependency model is too simple in current formulations

Example problem 3: Client misconfig

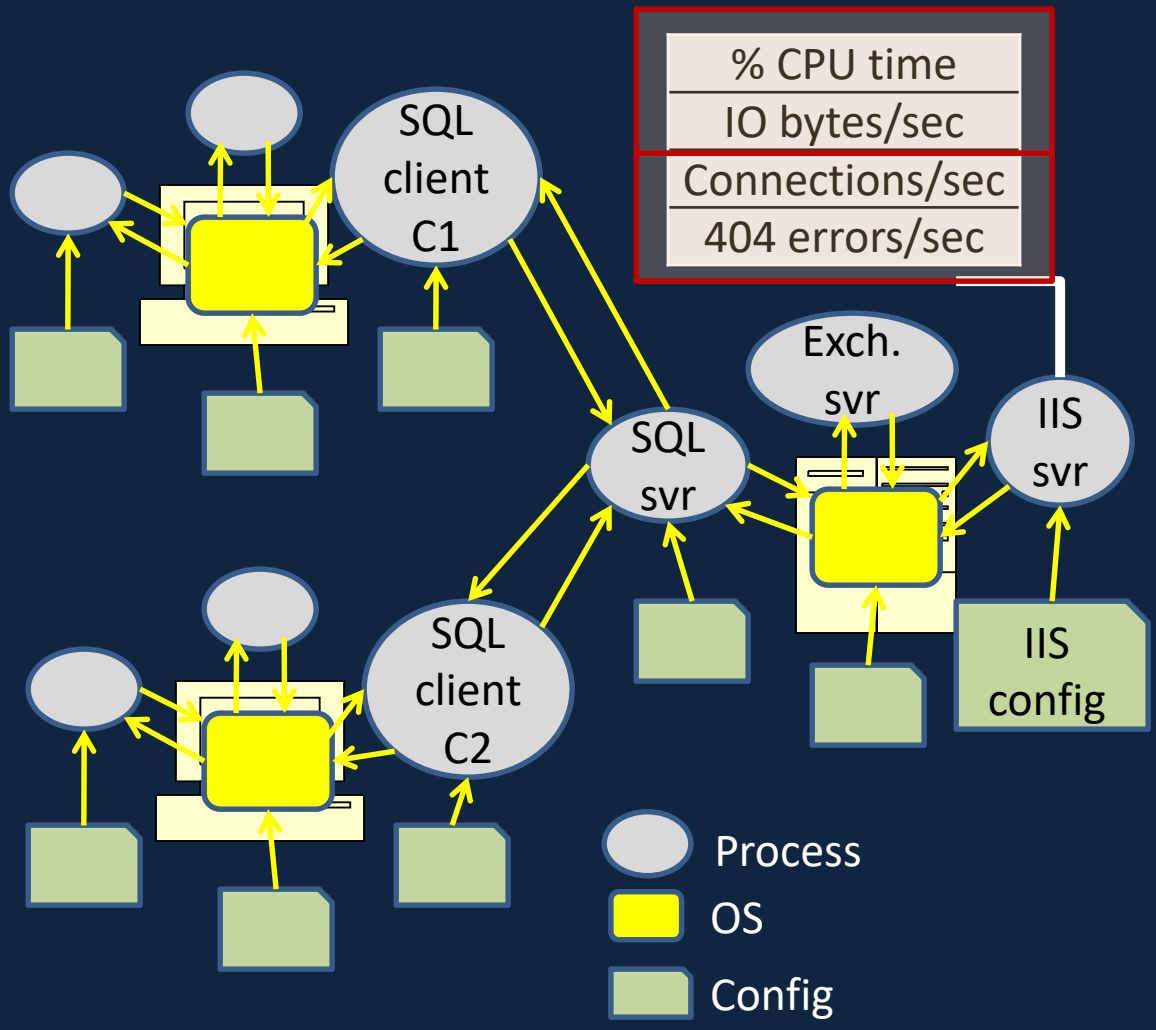


The failure model is too simple in current formulations

A formulation for detailed diagnosis

Dependency graph of fine-grained components

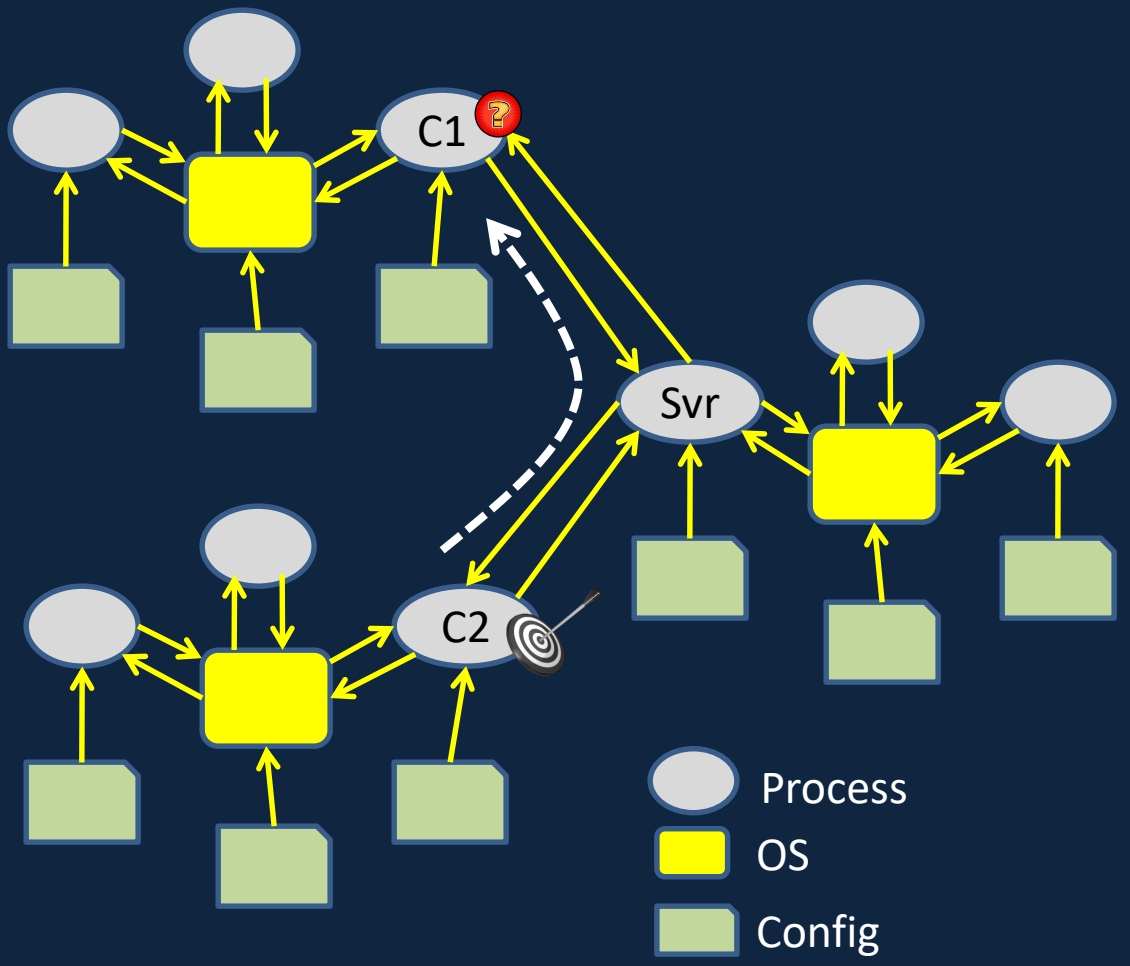
Component state is a multi-dimensional vector



The goal of diagnosis

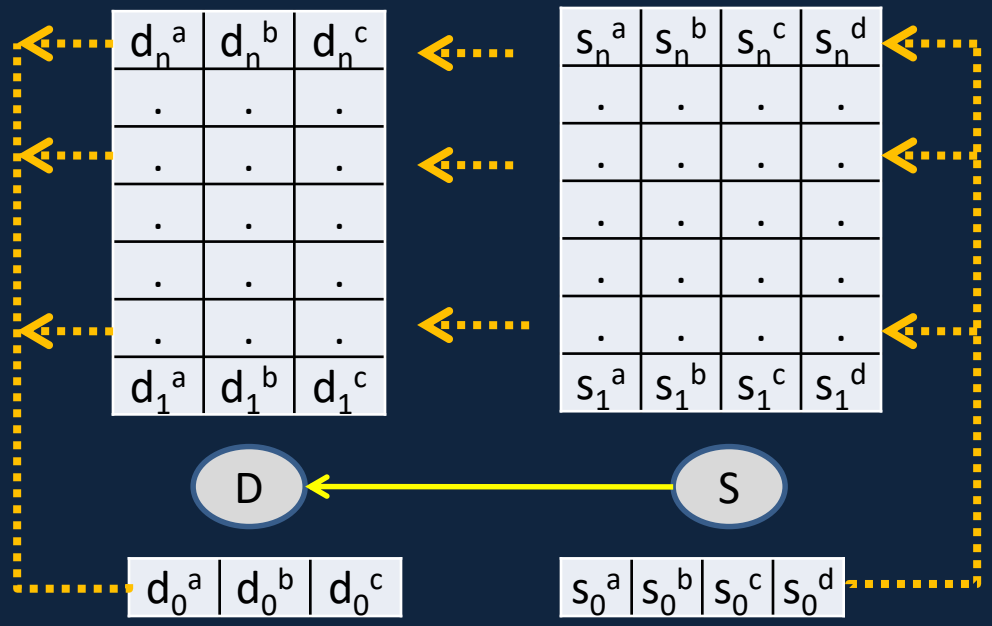
Identify likely culprits for components of interest

Without using semantics of state variables
➔ No application knowledge



Using joint historical behavior to estimate impact

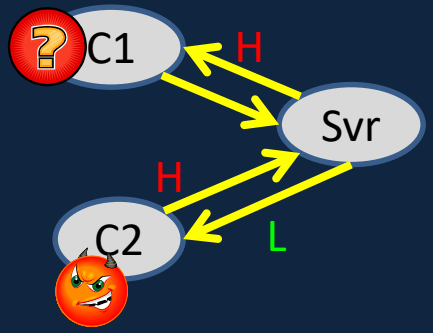
How "similar" on average states of D are at those times



Identify time periods when state of S was "similar"

Request rate (low)
Response time (high)

Request rate (high)
Response time (high)

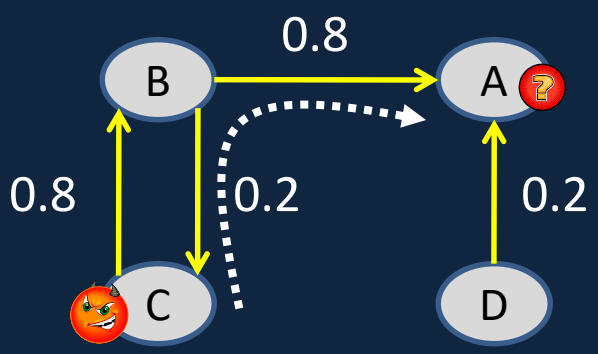


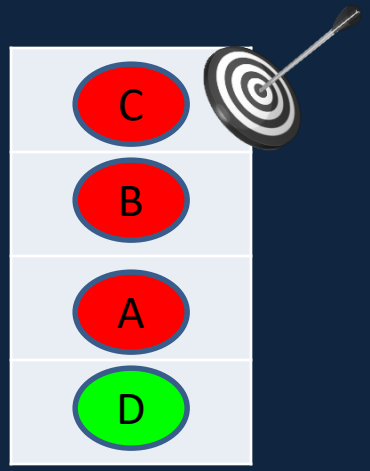
Request rate (high)

Robust impact estimation

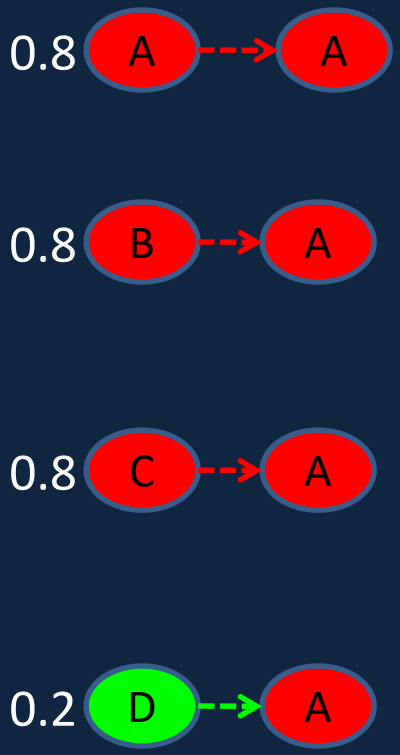
- Ignore state variables that represent redundant info
- Place higher weight on state variables likely related to fault being diagnosed
- Ignore state variables irrelevant to interaction with neighbor
- Account for aggregate relationships among state variables of neighboring components
- Account for disparate ranges of state variables

Ranking likely culprits

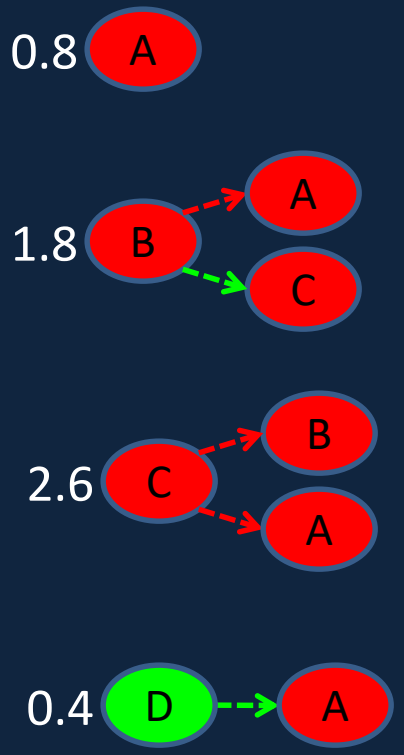




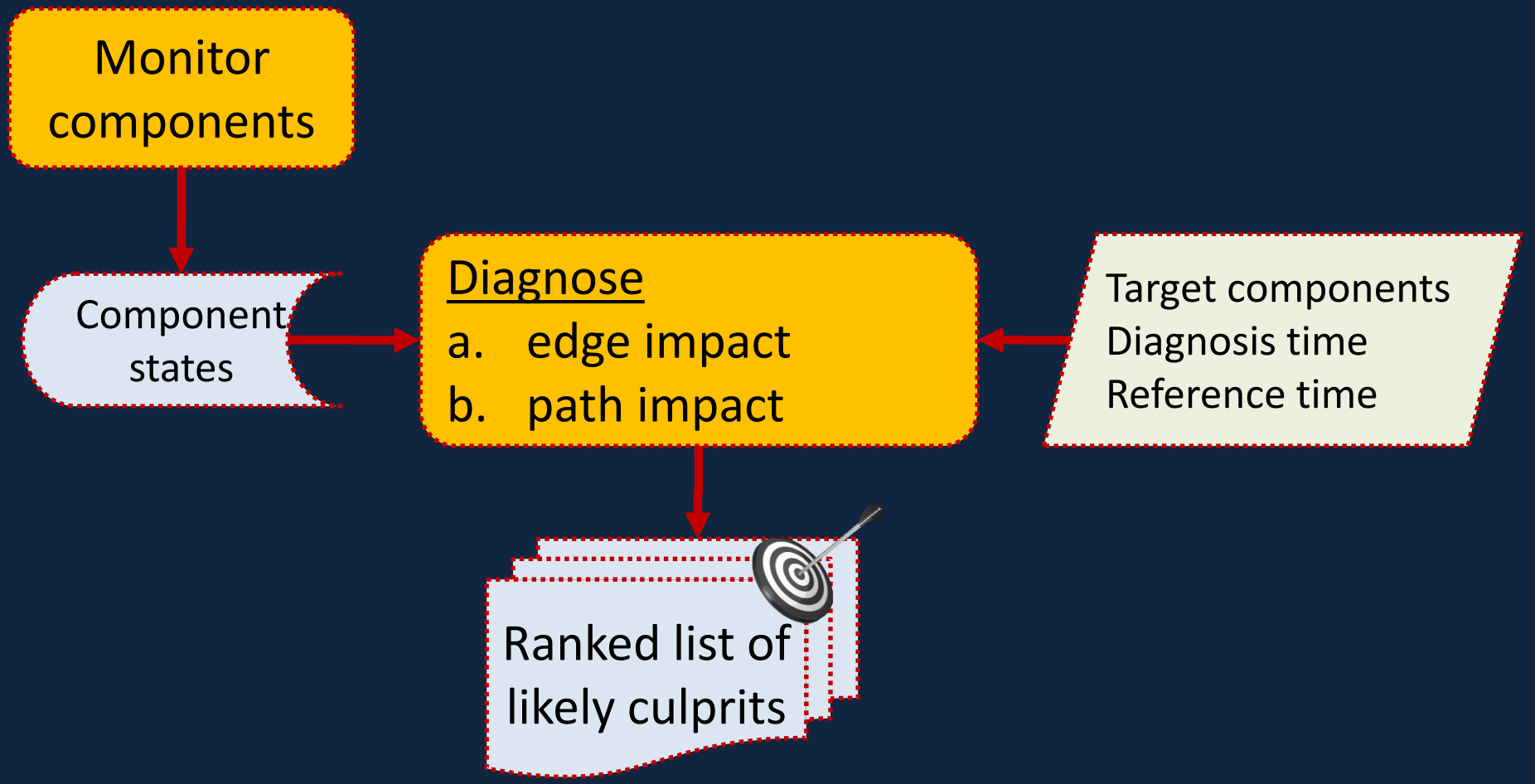
Path weight



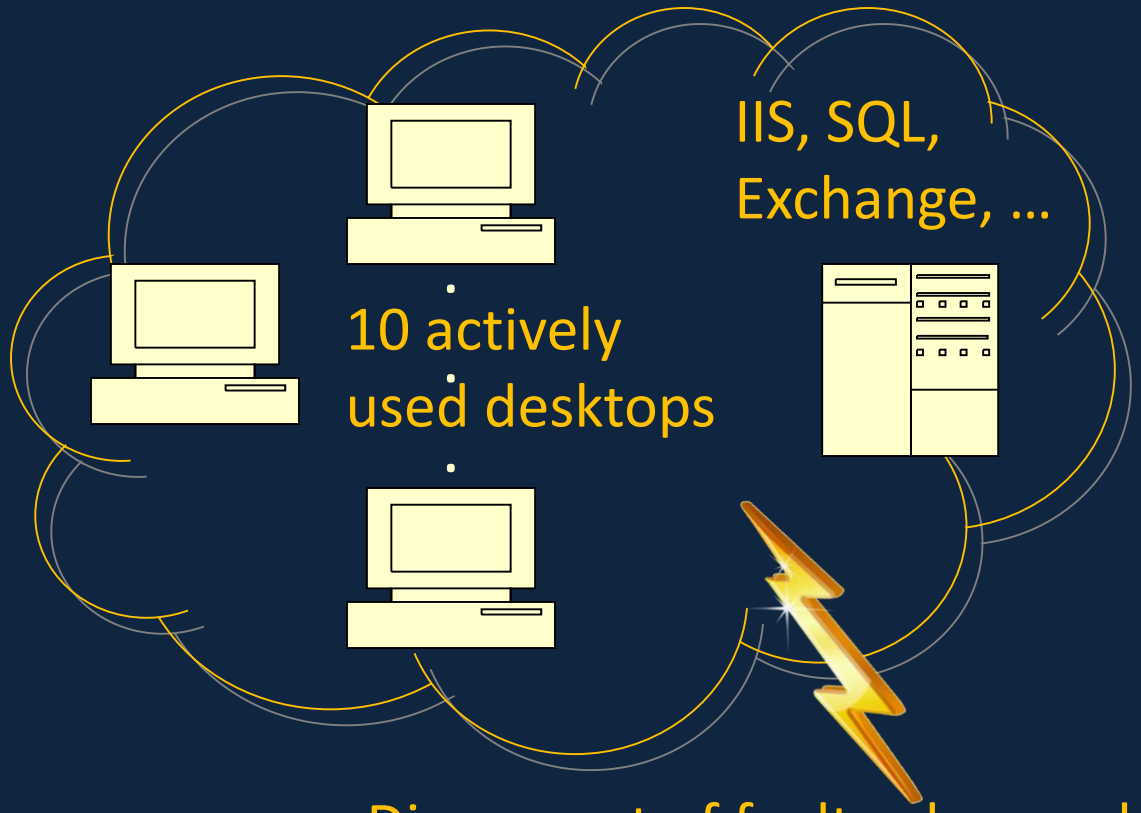
Global impact



Implementation of NetMedic



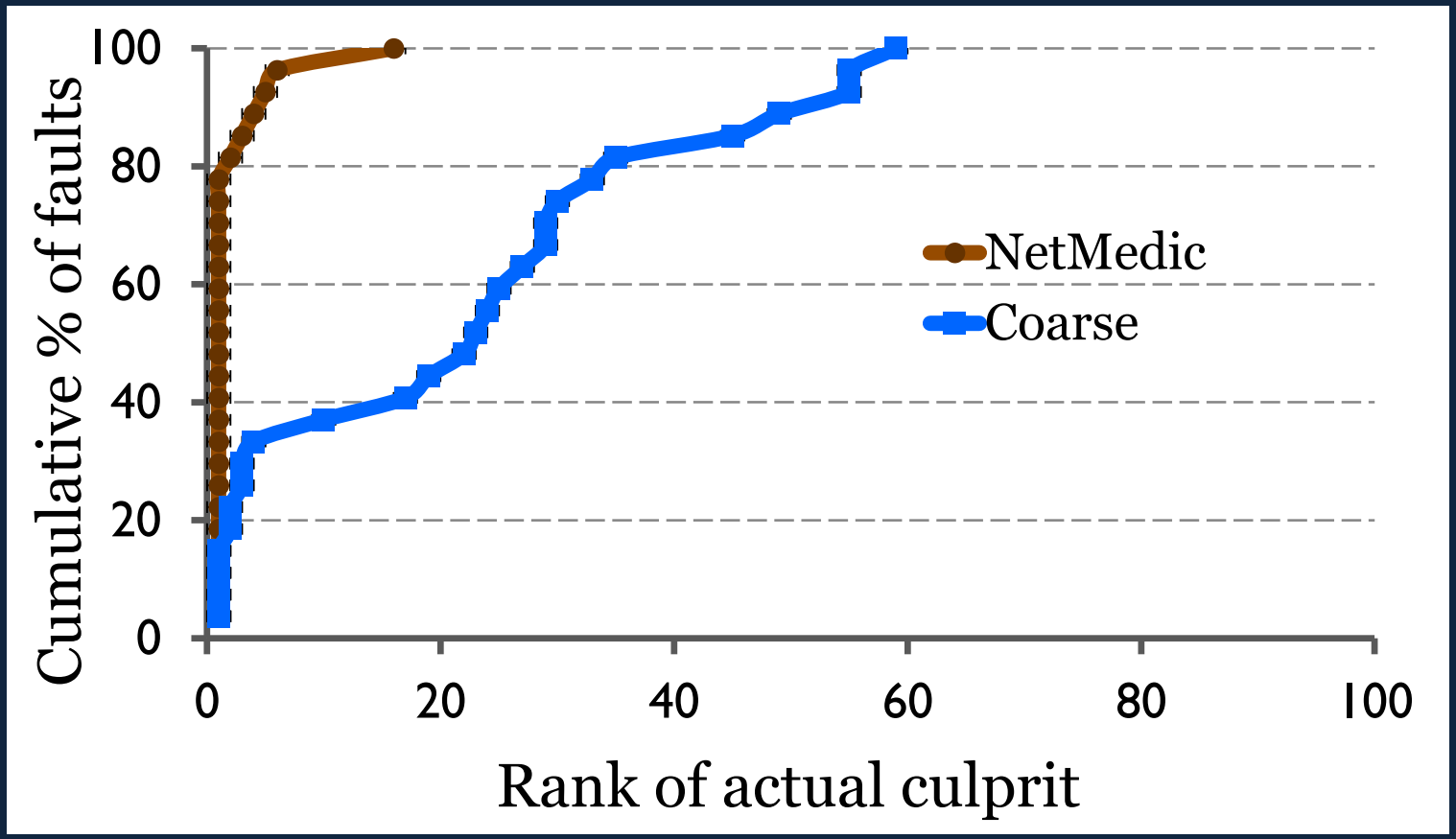
Evaluation setup



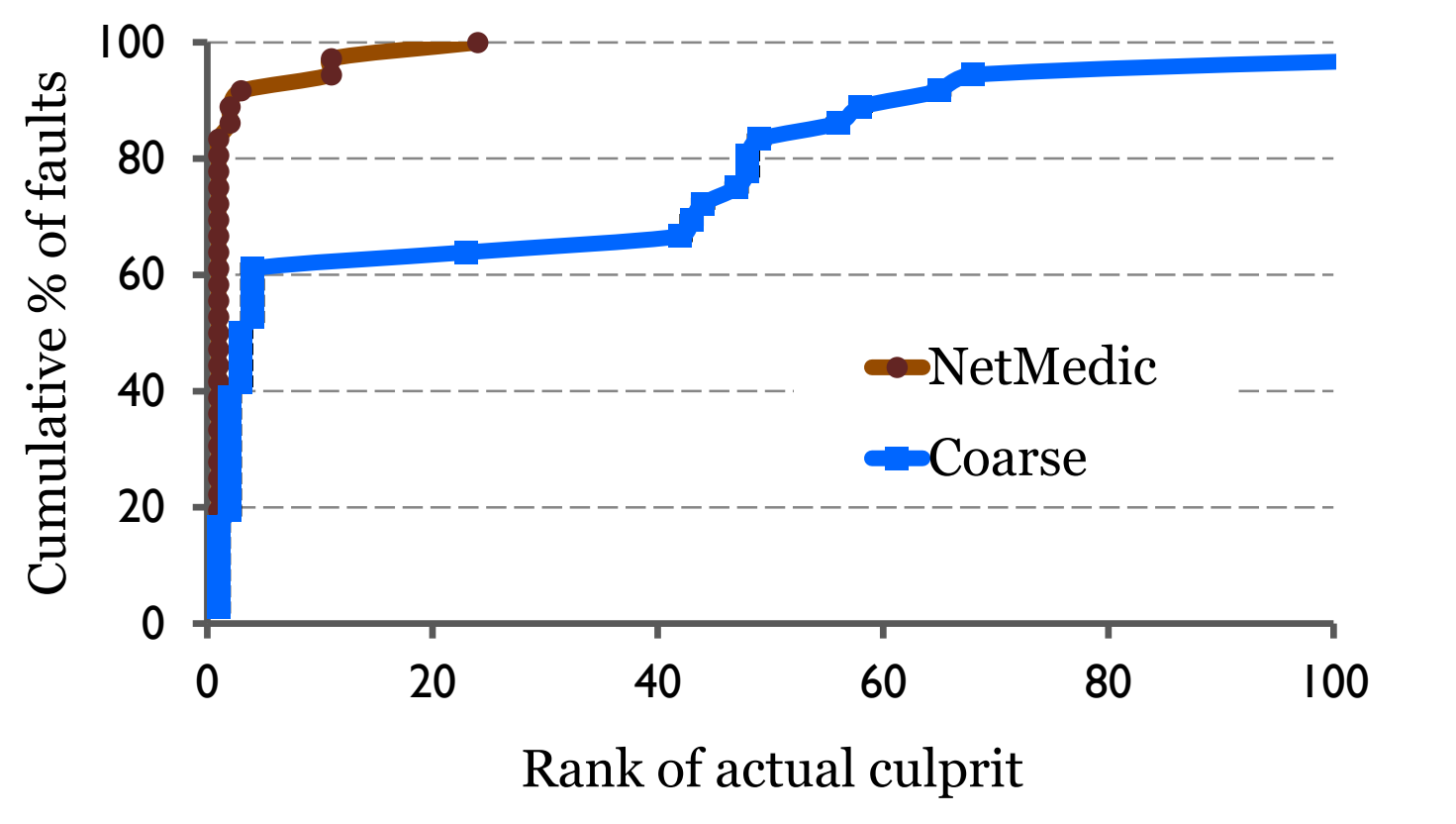
#components	~1000
#dimensions per component (avg)	35

Diverse set of faults observed in the logs

NetMedic assigns low ranks to actual culprits



NetMedic handles concurrent faults well



2 simultaneous faults

Other empirical results

Netmedic needs a modest amount (~60 mins) of history

The key to effectiveness is correctly identifying many low impact edges

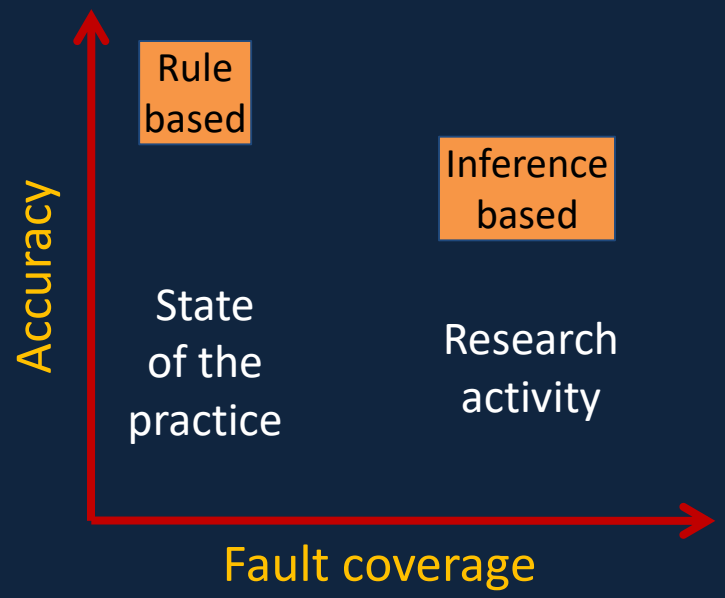
It compares favorably with a method that understands variable semantics

Unleashing (systems like) NetMedic on admins

How to present the analysis results?

- Need human verification

(Fundamental?) trade-off between coverage and accuracy



The understandability challenge

Admins should be able to verify the correctness of the analysis

- Identify culprits themselves if analysis is incorrect

Two sub-problems at the intersection with HCI

- Visualizing complex analysis (NetClinic)
- Intuitiveness of analysis (ongoing work)

NetClinic: Visualizing diagnostic analysis

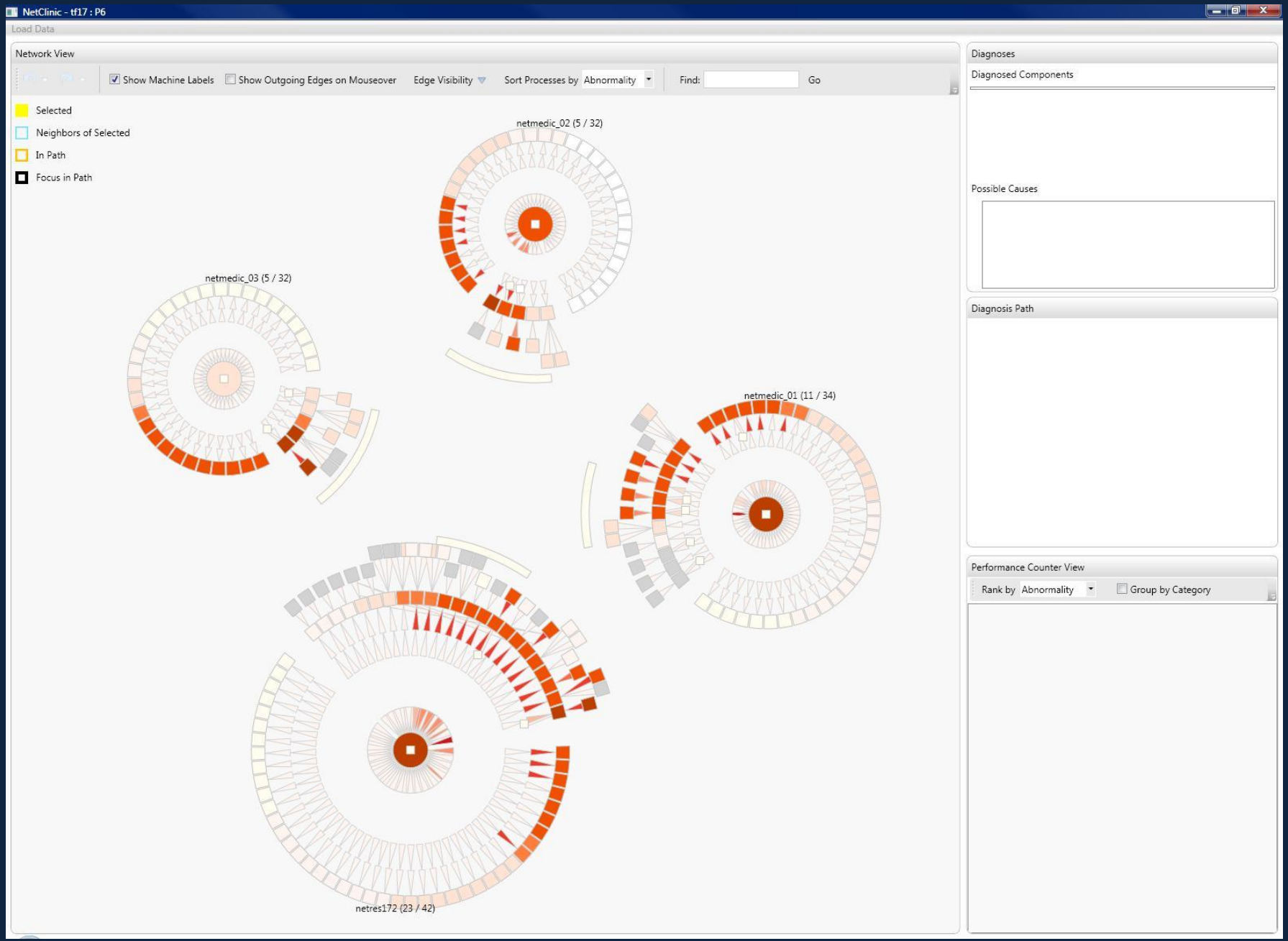
Underlying assumption: Admins can verify analysis if information is presented appropriately

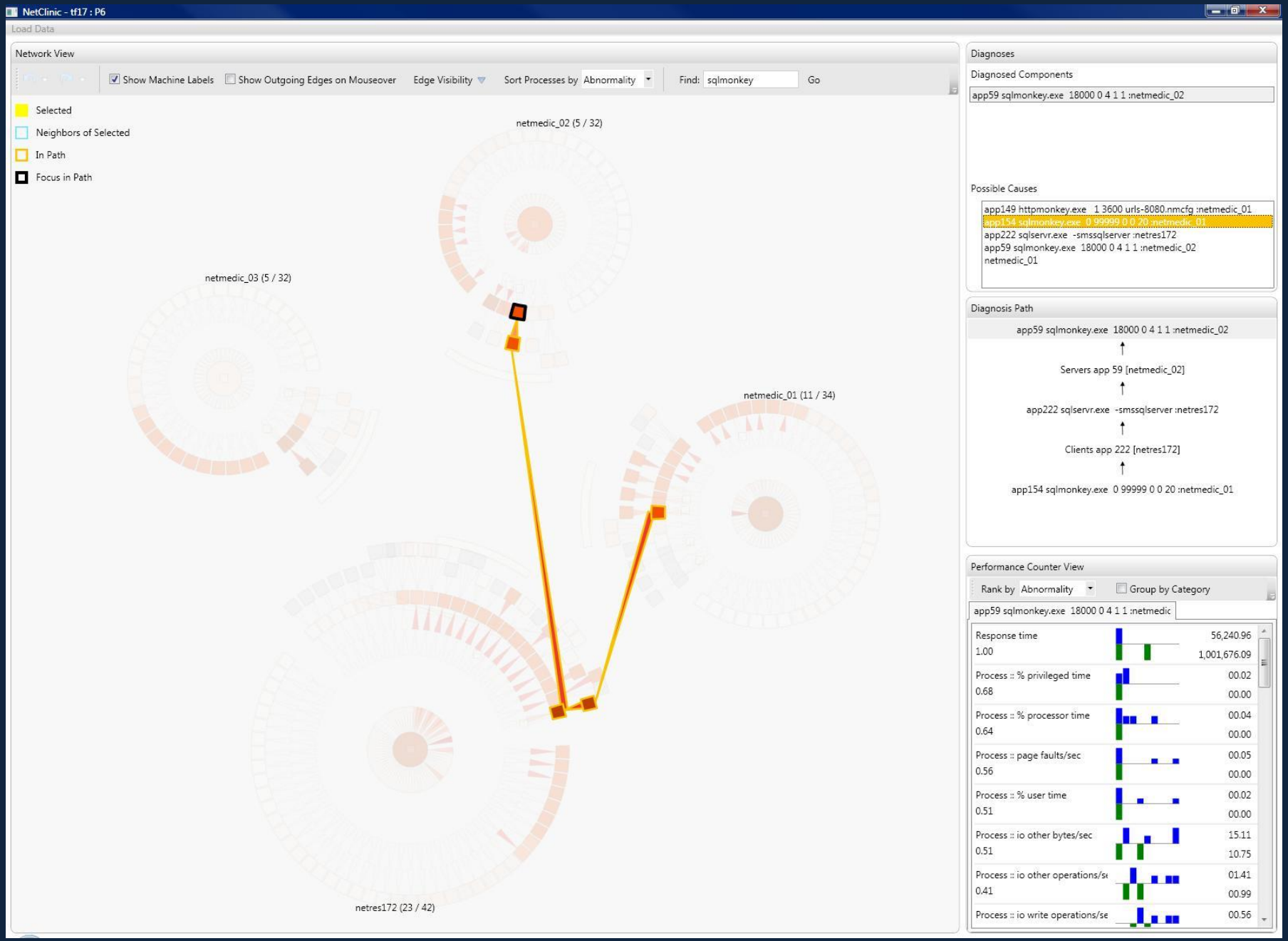
- They have expert, out-of-band information

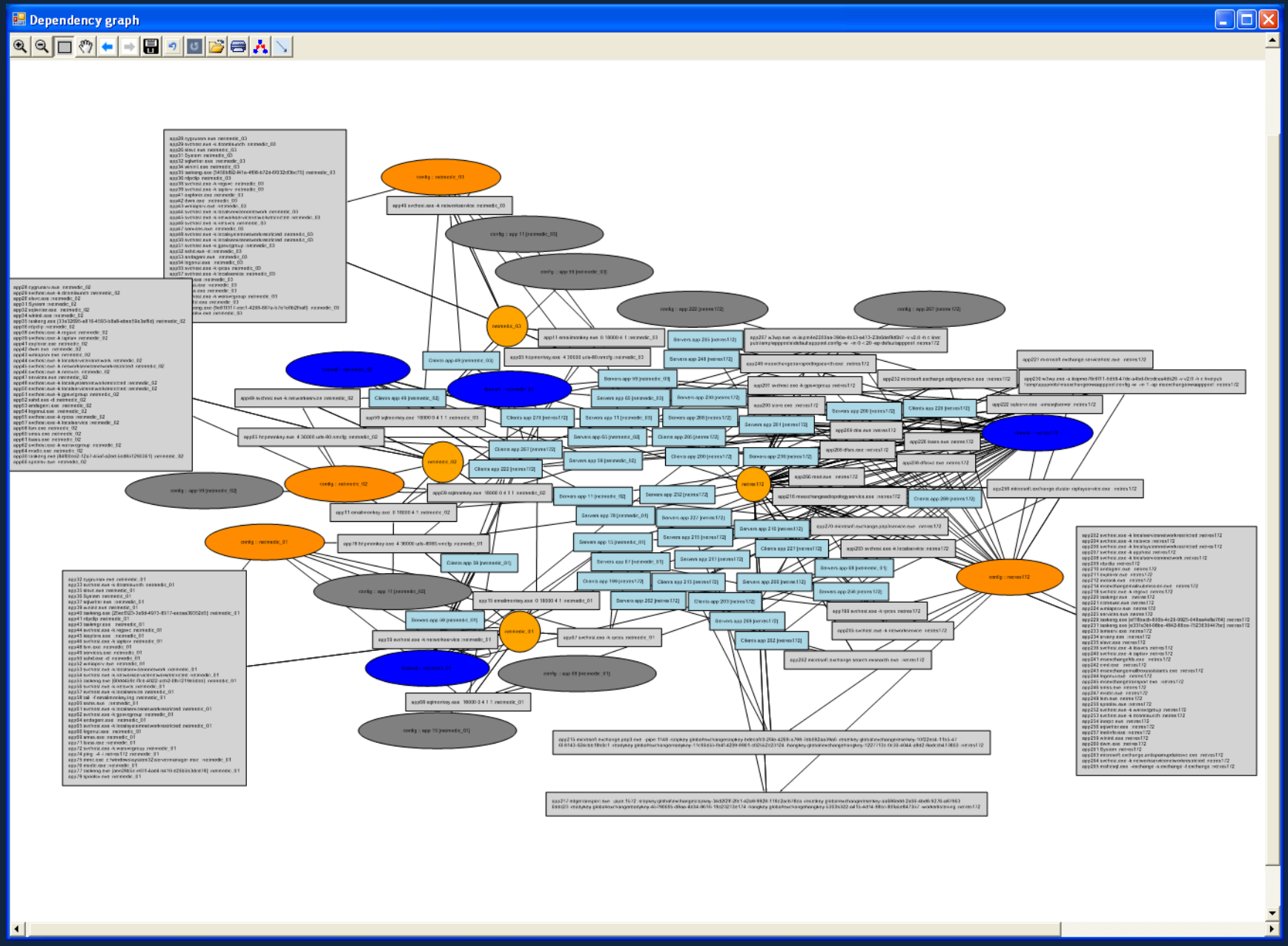
Views diagnosis as multi-level analysis

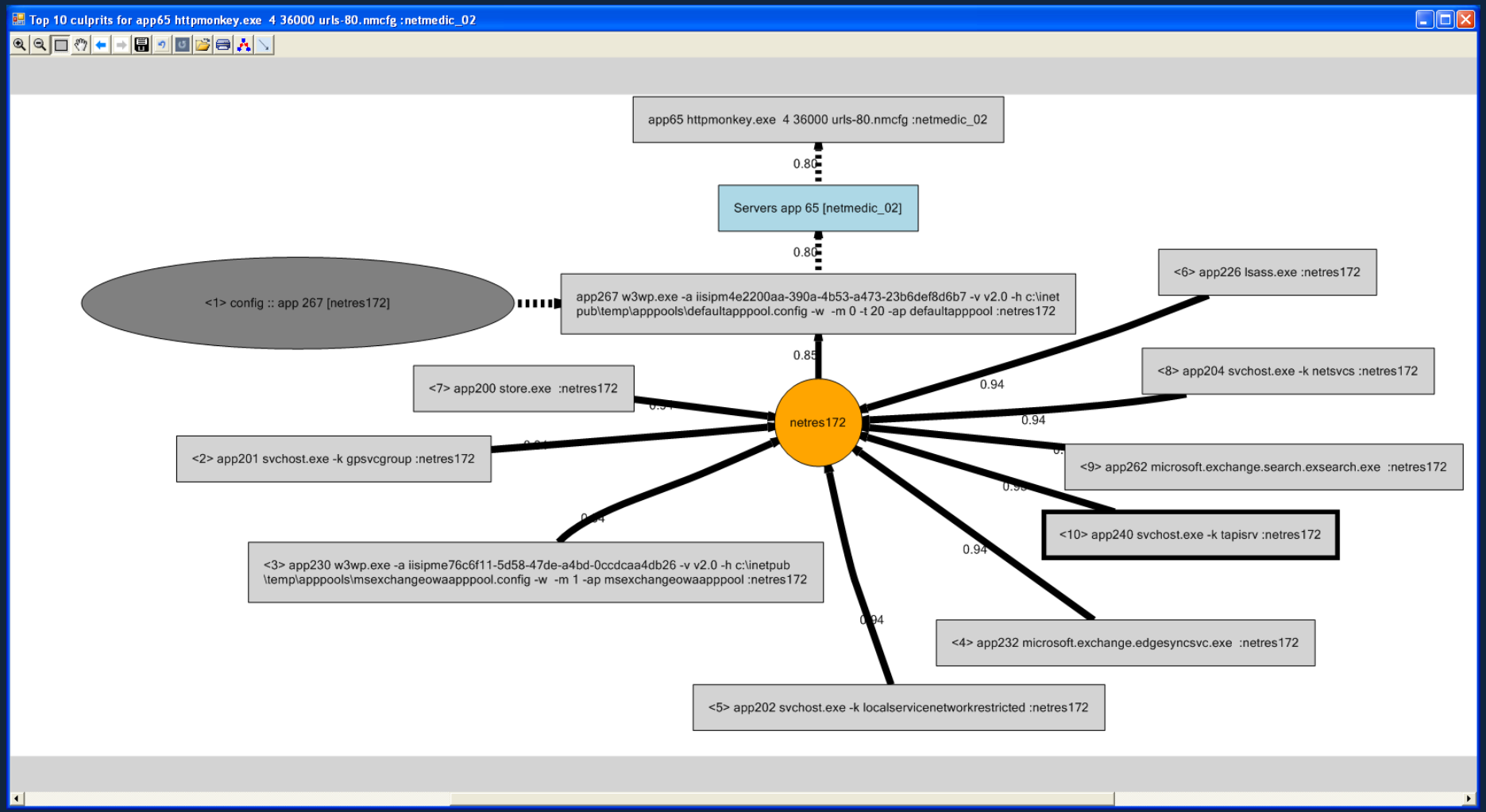
Makes results at all levels accessible on top of a semantic graph layout

Allows top-down and bottom-up navigation across levels while retaining context


NetClinic - tf17 : P6
Load Data
Network View
Show Machine Labels
Show Outgoing Edges on Mouseover
Edge Visibility
Sort Processes by Abnormality
Find:
Go
Selected
Neighbors of Selected
In Path
Focus in Path
netmedic_02 (5 / 32)
netmedic_03 (5 / 32)
netmedic_01 (11 / 34)
netres172 (23 / 42)
Diagnoses
Diagnosed Components
Possible Causes
Diagnosis Path
Performance Counter View
Rank by Abnormality
Group by Category


NetClinic - tf17 : P6
Load Data
Network View
Show Machine Labels
Show Outgoing Edges on Mouseover
Edge Visibility
Sort Processes by Abnormality
Find: sqlmonkey
Go
Selected
Neighbors of Selected
In Path
Focus in Path
netmedic_02 (5 / 32)
netmedic_03 (5 / 32)
netmedic_01 (11 / 34)
netres172 (23 / 42)
Diagnoses
Diagnosed Components
app59 sqlmonkey.exe 18000 0 4 1 1 :netmedic_02
Possible Causes
app149 httpmonkey.exe 1 3600 urls-8080.nmcfg :netmedic_01
app154 sqlmonkey.exe 0 99999 0 0 20 :netmedic_01
app222 sqlservr.exe -smssqlserver :netres172
app59 sqlmonkey.exe 18000 0 4 1 1 :netmedic_02
netmedic_01
Diagnosis Path
app59 sqlmonkey.exe 18000 0 4 1 1 :netmedic_02
Servers app 59 [netmedic_02]
app222 sqlservr.exe -smssqlserver :netres172
Clients app 222 [netres172]
app154 sqlmonkey.exe 0 99999 0 0 20 :netmedic_01
Performance Counter View
Rank by Abnormality
Group by Category
app59 sqlmonkey.exe 18000 0 4 1 1 :netmedic
Response time
1.00
56,240.96
1,001,676.09
Process :: % privileged time
0.68
00.02
00.00
Process :: % processor time
0.64
00.04
00.00
Process :: page faults/sec
0.56
00.05
00.00
Process :: % user time
0.51
00.02
00.00
Process :: io other bytes/sec
0.51
15.11
10.75
Process :: io other operations/se
0.41
01.41
00.99
Process :: io write operations/se
00.56


Dependency graph


Top 10 culprits for app65 httpmonkey.exe 4 36000 urls-80.nmcfg :netmedic_02
app65 httpmonkey.exe 4 36000 urls-80.nmcfg :netmedic_02
0.80
Servers app 65 [netmedic_02]
0.80
<1> config :: app 267 [netres172]
app267 w3wp.exe -a iisipm4e2200aa-390a-4b53-a473-23b6def8d6b7 -v v2.0 -h c:\inetpub\temp\apppools\defaultapppool.config -w -m 0 -t 20 -ap defaultapppool :netres172
0.85
<6> app226 lsass.exe :netres172
<8> app204 svchost.exe -k netsvcs :netres172
<7> app200 store.exe :netres172
0.94
0.94
netres172
<2> app201 svchost.exe -k gpsvcgroup :netres172
<9> app262 microsoft.exchange.search.exsearch.exe :netres172
<10> app240 svchost.exe -k tapisrv :netres172
<3> app230 w3wp.exe -a iisipme76c6f11-5d58-47de-a4bd-0ccdcaa4db26 -v v2.0 -h c:\inetpub\temp\apppools\msexchangeowaapppool.config -w -m 1 -ap msexchangeowaapppool :netres172
0.94
<4> app232 microsoft.exchange.edgesyncsvc.exe :netres172
<5> app202 svchost.exe -k localservicenetworkrestricted :netres172

NetClinic user study

11 participants with knowledge of computer networks but not of *NetMedic*

Given 3 diagnostic tasks each after training

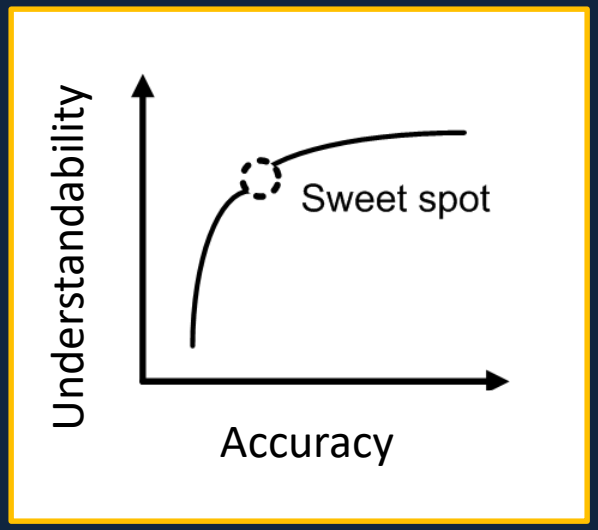
- 88% task completion rate

Uncovered a rich mix of user strategies that the visualization must support

Intuitiveness of analysis

What if you could modify the analysis itself to make it more accessible to humans?

- Counters the tendency to “optimize” for incremental gains in accuracy



Intuitiveness of analysis (2)

Goal: Go from mechanical measures to more human centric measures

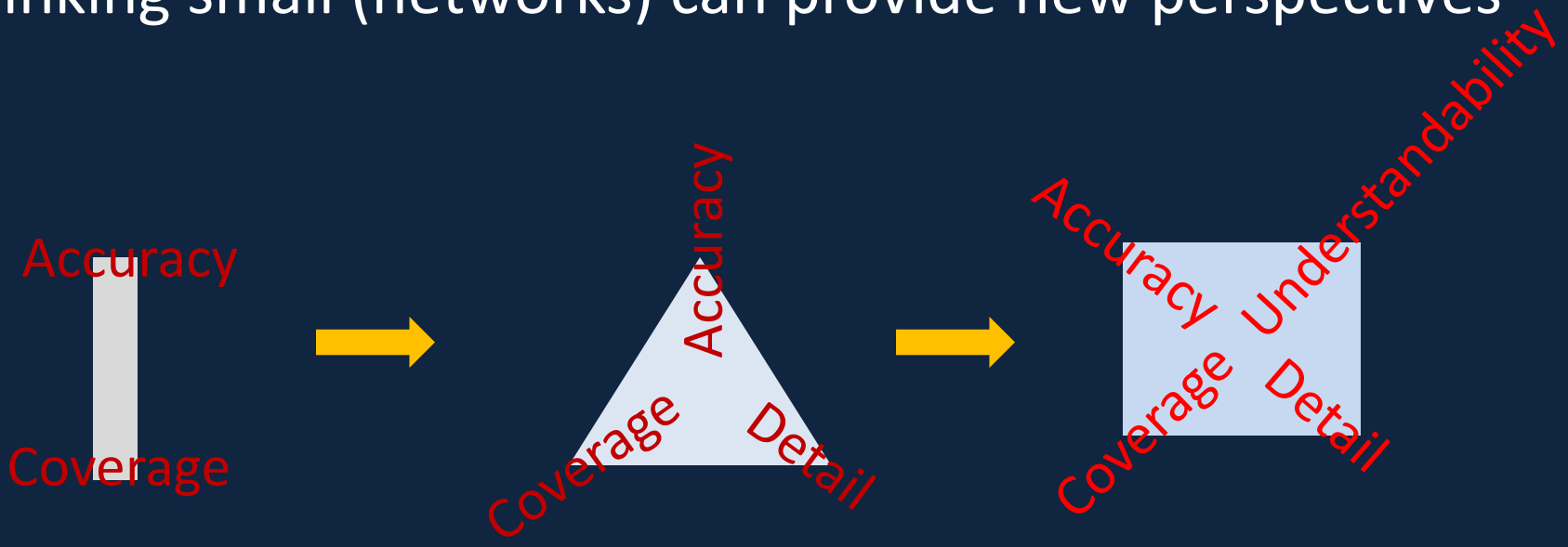
- Example: MoS measure for VoIP

Factors to consider

- What information is used? E.g., Local vs. global
- What operations are used? E.g., Arithmetic vs. geometric means

Conclusions

Thinking small (networks) can provide new perspectives



NetMedic enables detailed diagnosis in enterprise networks w/o application knowledge

NetClinic enables admins to understand and verify complex diagnostic analyses