An Empirical Study on Configuration Errors in Commercial and Open Source Systems

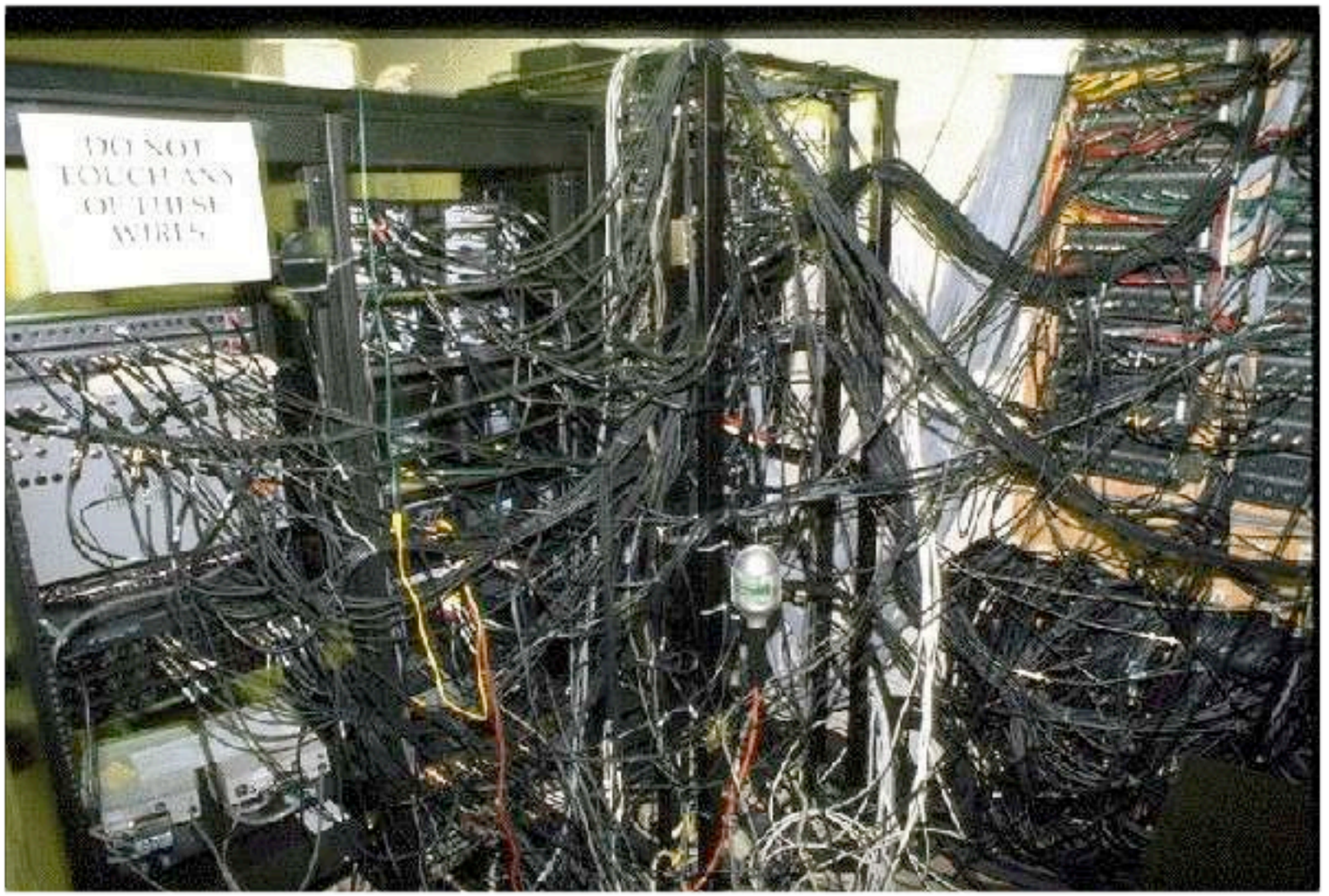
Zuoning Yin, Xiao ma,
Jing Zheng, Yuanyuan Zhou

University of California
At San Diego

Lakshmi N. Bairavasundaram,
Shankar Pasupathy

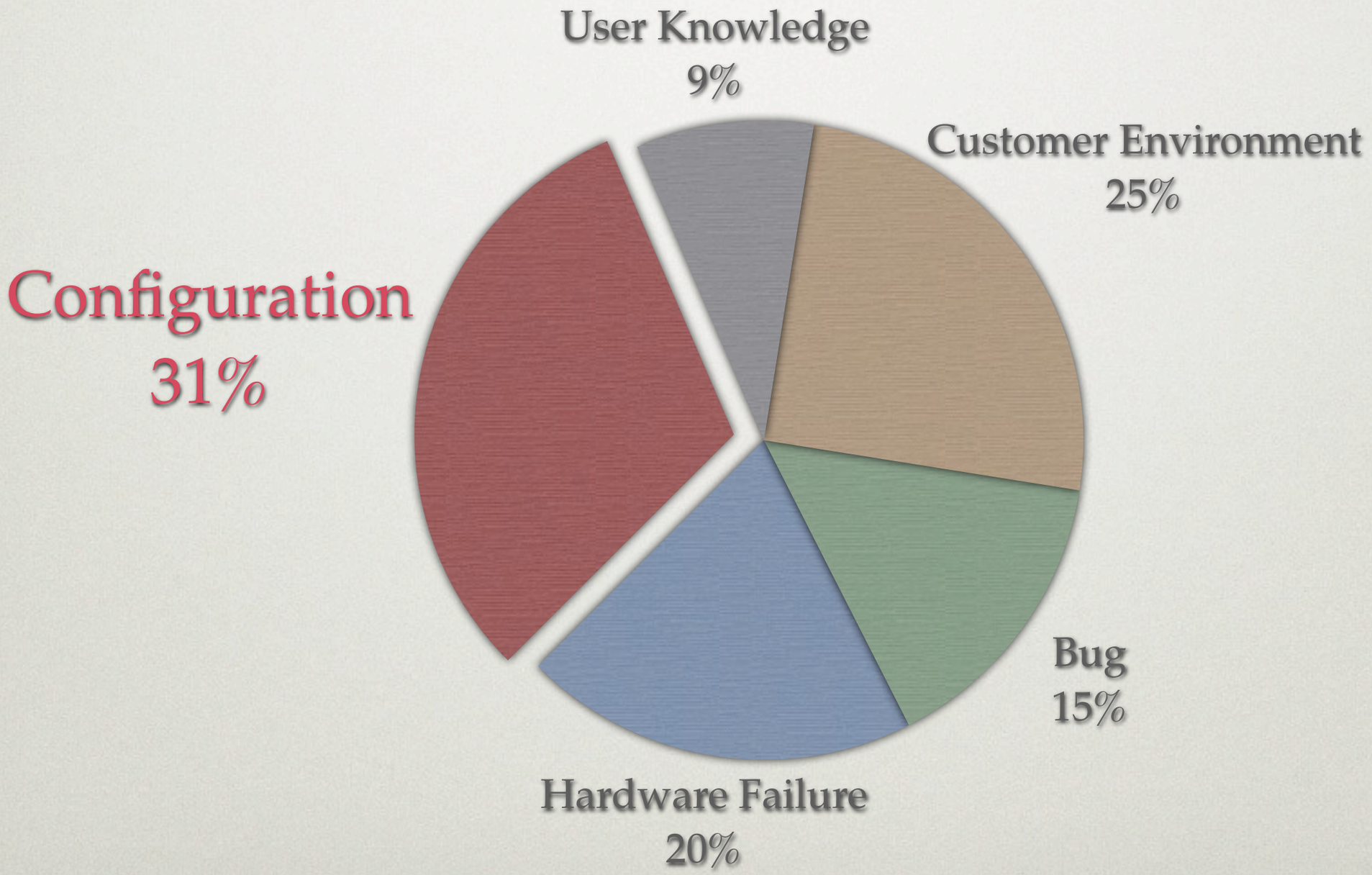
Netapp Inc.

Configuring computers is not easy



Configuring server systems is much harder

Root Causes of Customer Reported Issues



Who should take responsibility for configuration errors, users or developers?

- Developers do not think carefully when they design configuration interface
- Once an issues turns out to be a configuration error, developers move on

How to Reduce Configuration Errors?

- What kind of configuration errors do users make?
- Which types of configuration parameters are more error-prone?
- Which user actions may cause configuration errors?
- .....

We need to understand real-world configuration errors first

Objectives and Challenges

- Objectives
  - Understand the characteristics of real-world configuration errors
  - Reveal their implications to developers
- Challenges
  - Configuration errors are not recorded rigorously
  - Configuration errors are difficult to understand

Methodology

- Random sampling configuration errors
  - Choose resolved cases in recent 2 years
  - Ensure the sample size is big enough
  - Calculate statistical error
- Categorizing errors with best effort
  - Cross-validation among co-authors
  - Help from developers

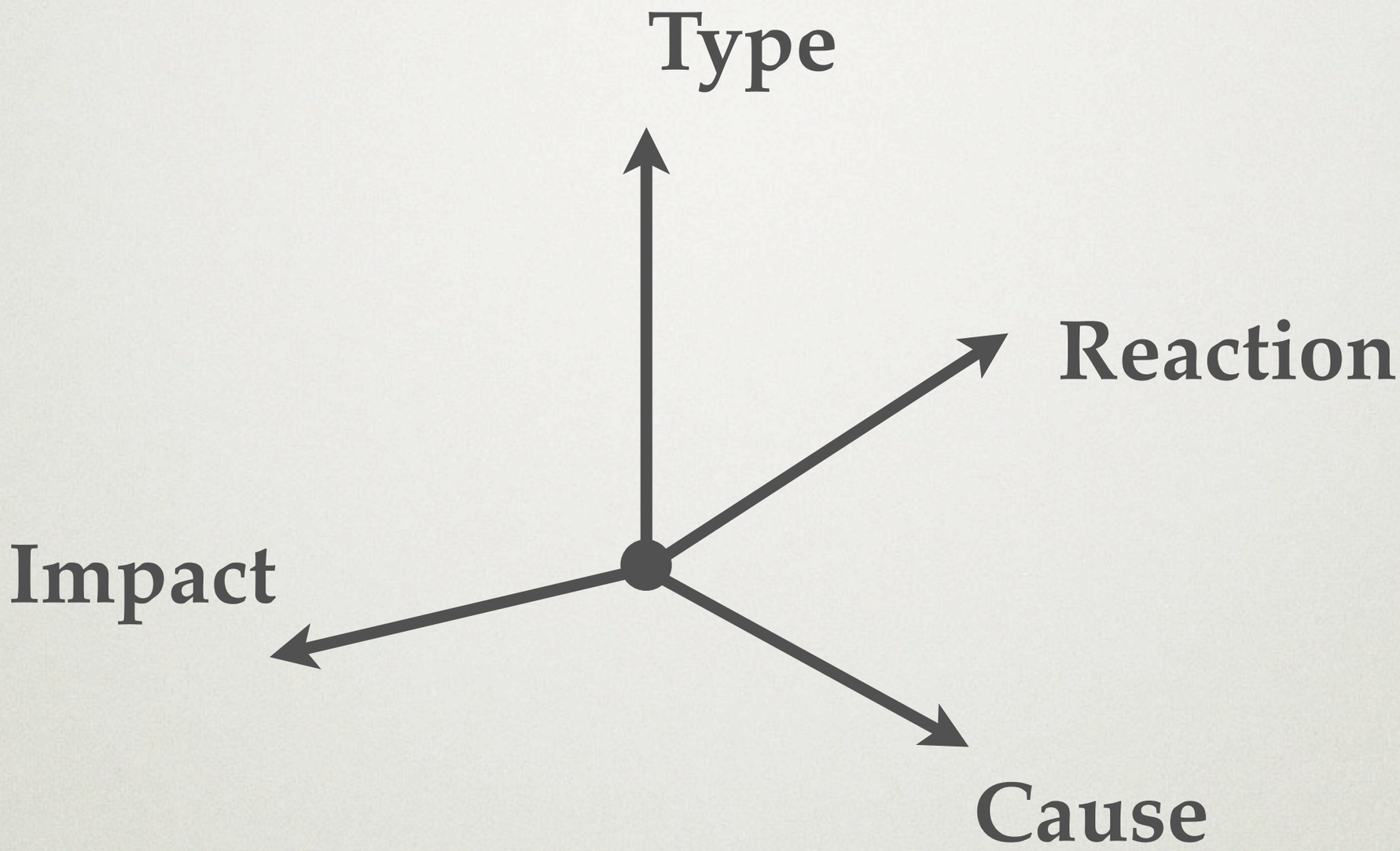
Data Source

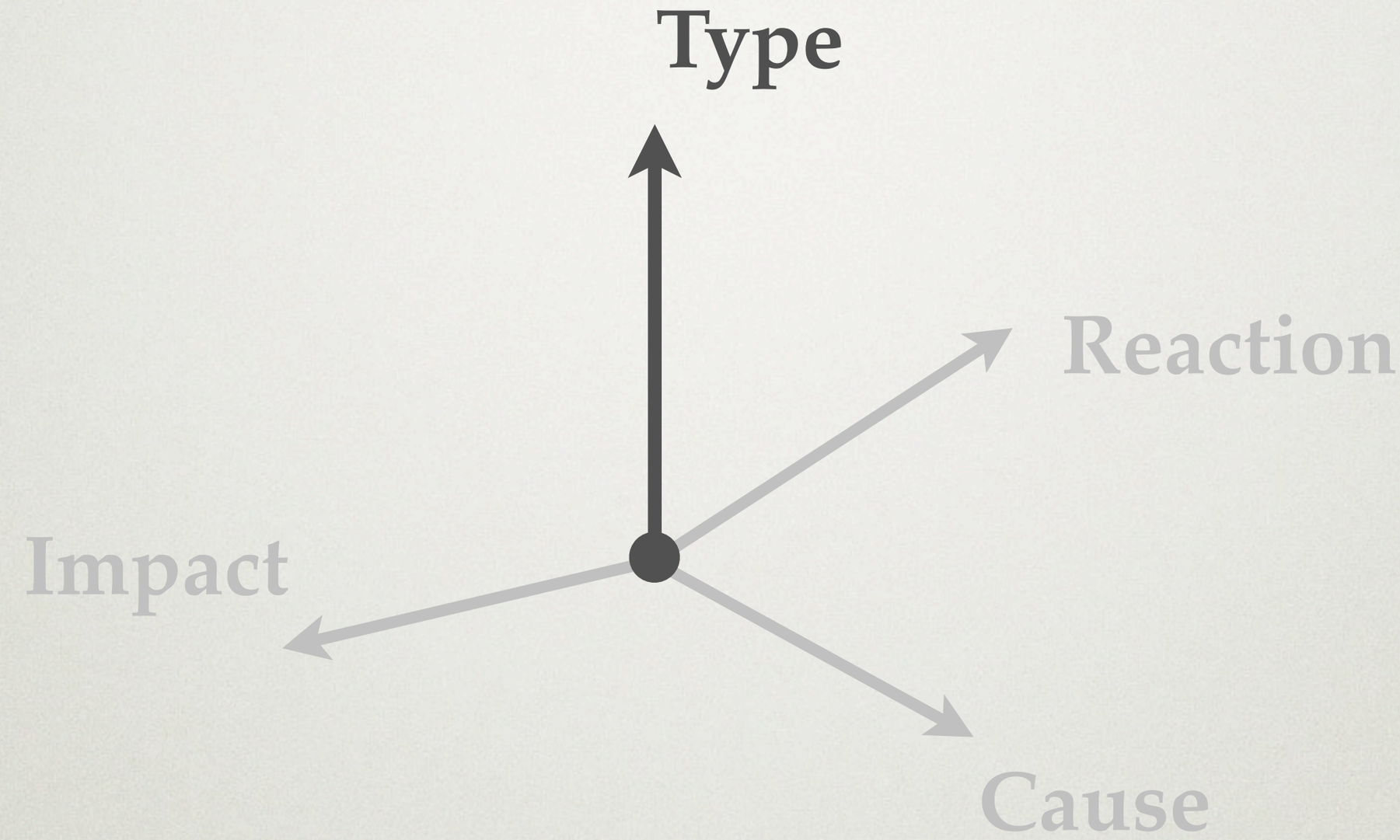
System		Number of Sampled Errors
Commercial	COMP-A	**309**
Open Source	CentOS	**60**
	MySQL	**55**
	Apache	**60**
	Open LDAP	**62**
Total		**546**

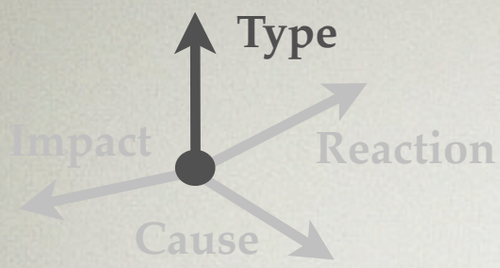
Limitation

- We study only user-reported errors
  - Configuration errors may be resolved in other means

- We focus on server-side systems
  - Other types of systems may have different characteristics


Type
Reaction
Impact
Cause


Type
Reaction
Impact
Cause

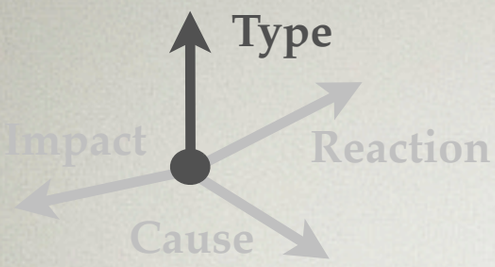


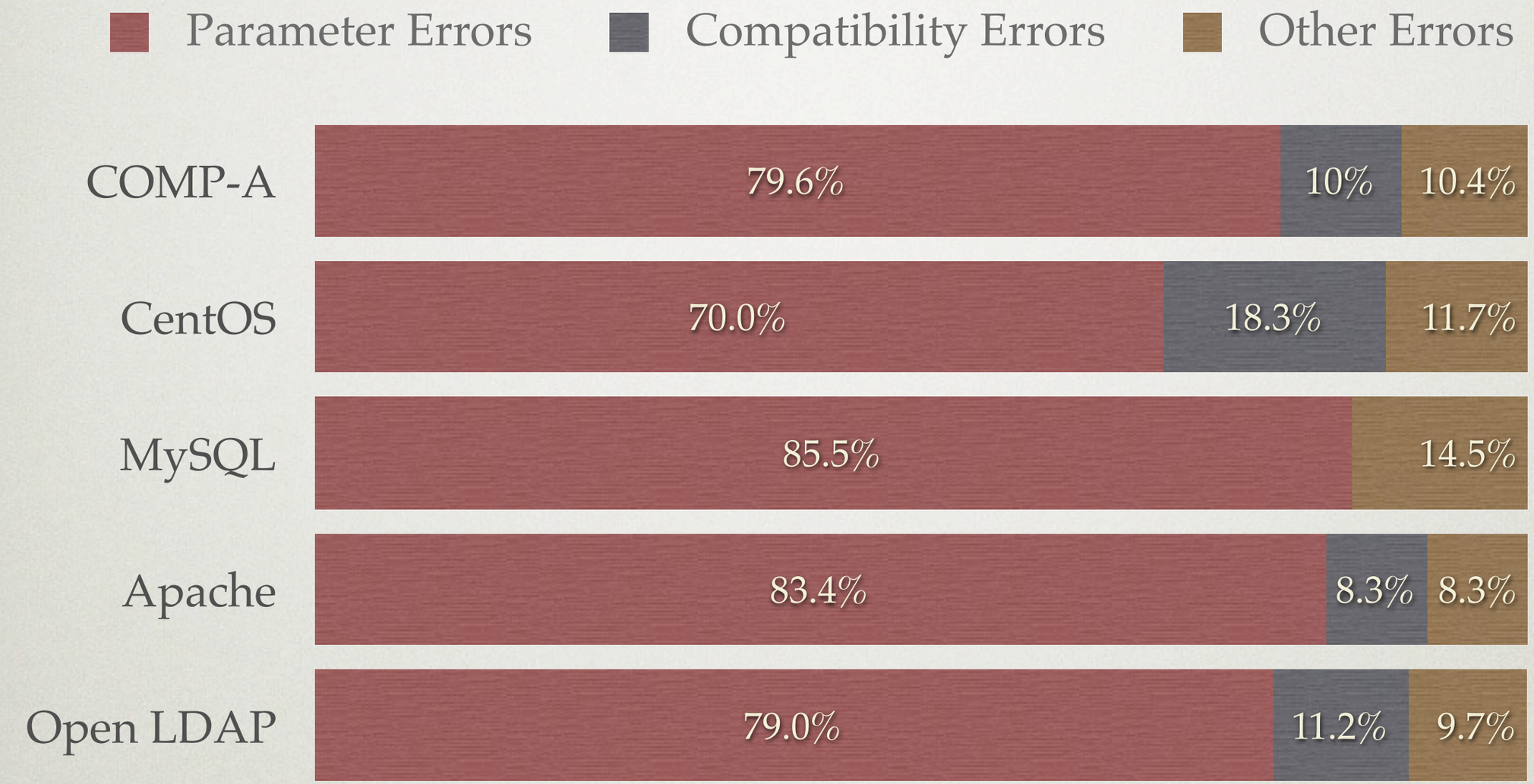
Software Configuration Errors

Parameter Errors

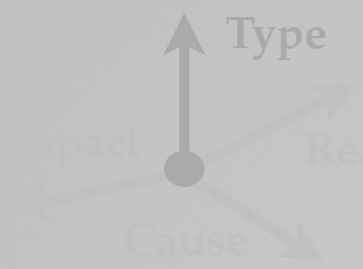
Compatibility Errors

Other Errors





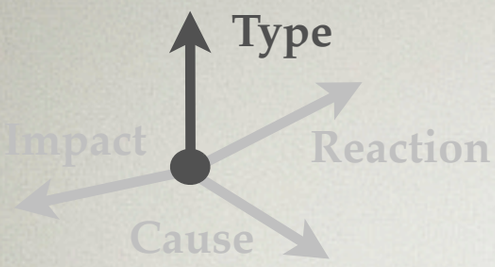
Parameter errors dominate

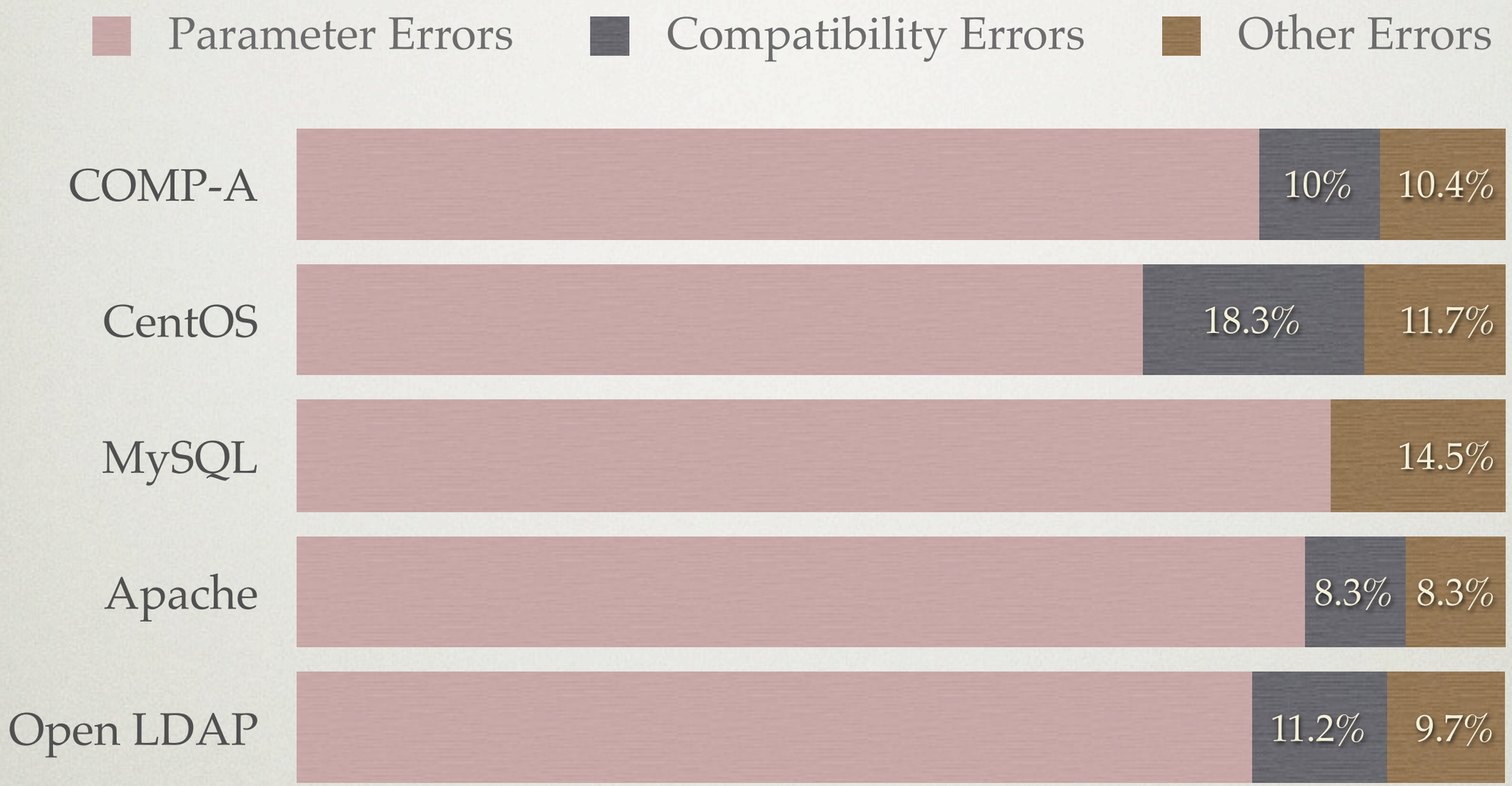


Parameter Errors Compatibility Errors Other Errors

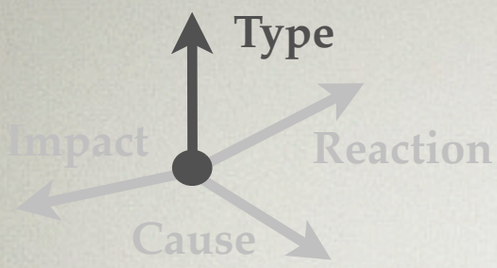
- Systems should expose as few configuration parameters as possible
- Automatic configuration is preferred

Parameter errors dominate





Other types of configuration errors are not negligible



Parameter Errors

COMP-A

```
InitiatorName: iqn_DEV_domain
```

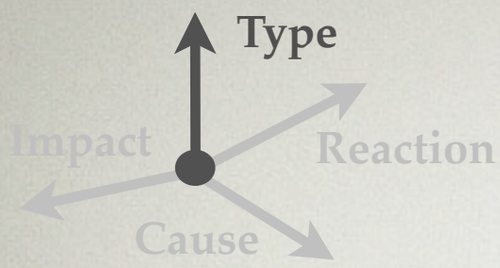
Lower-case only value

Error!

MySQL

```
AutoCommit = True
```

“True” value may affect performance



Parameter Errors

Illegal Parameters

COMP-A

```
InitiatorName: iqn_DEV_domain
```

Lower-case only value

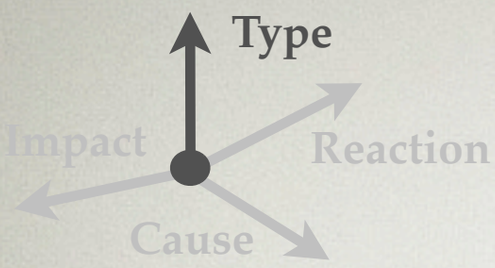
Error!

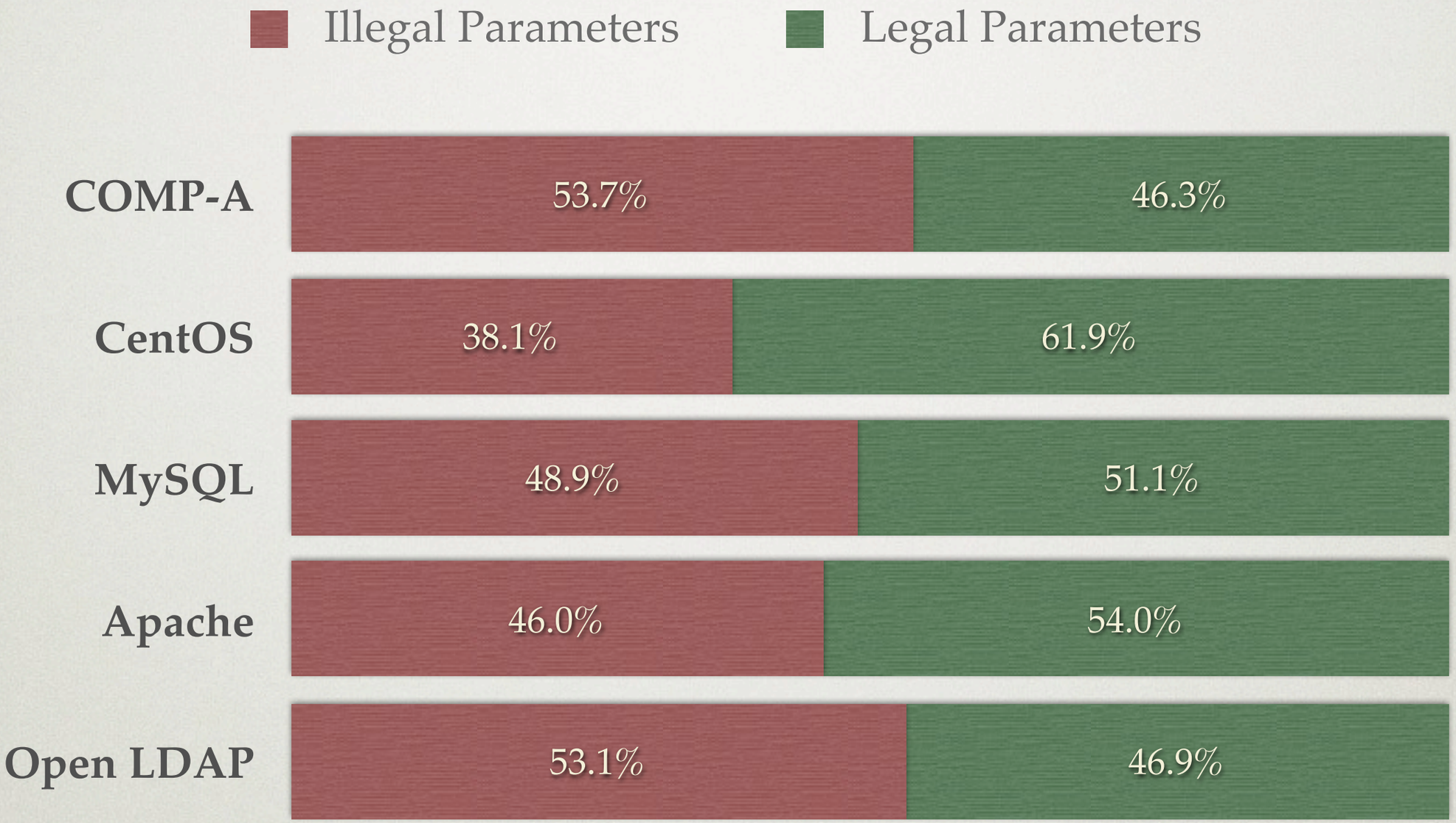
Legal Parameters

MySQL

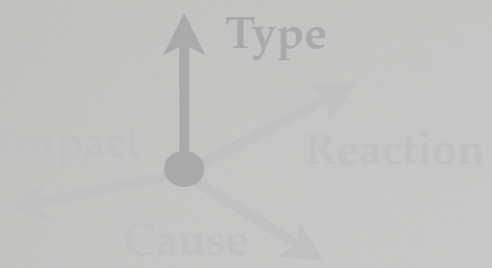
```
AutoCommit = True
```

“True” value may affect performance



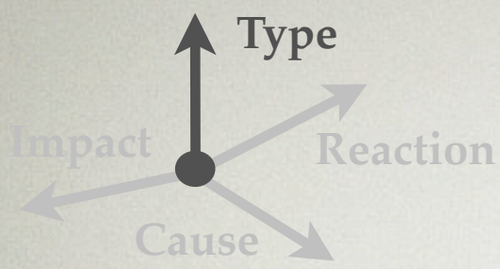


Illegal and legal parameters have similar contribution to parameter errors



- Illegal parameters are relatively easy to detect
- About half of parameter errors involve illegal parameters (“good” news!)

Illegal and legal parameters have similar contribution to parameter errors



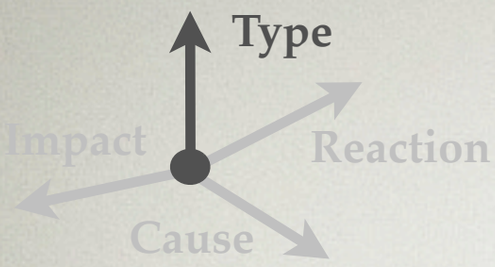
Illegal Parameters

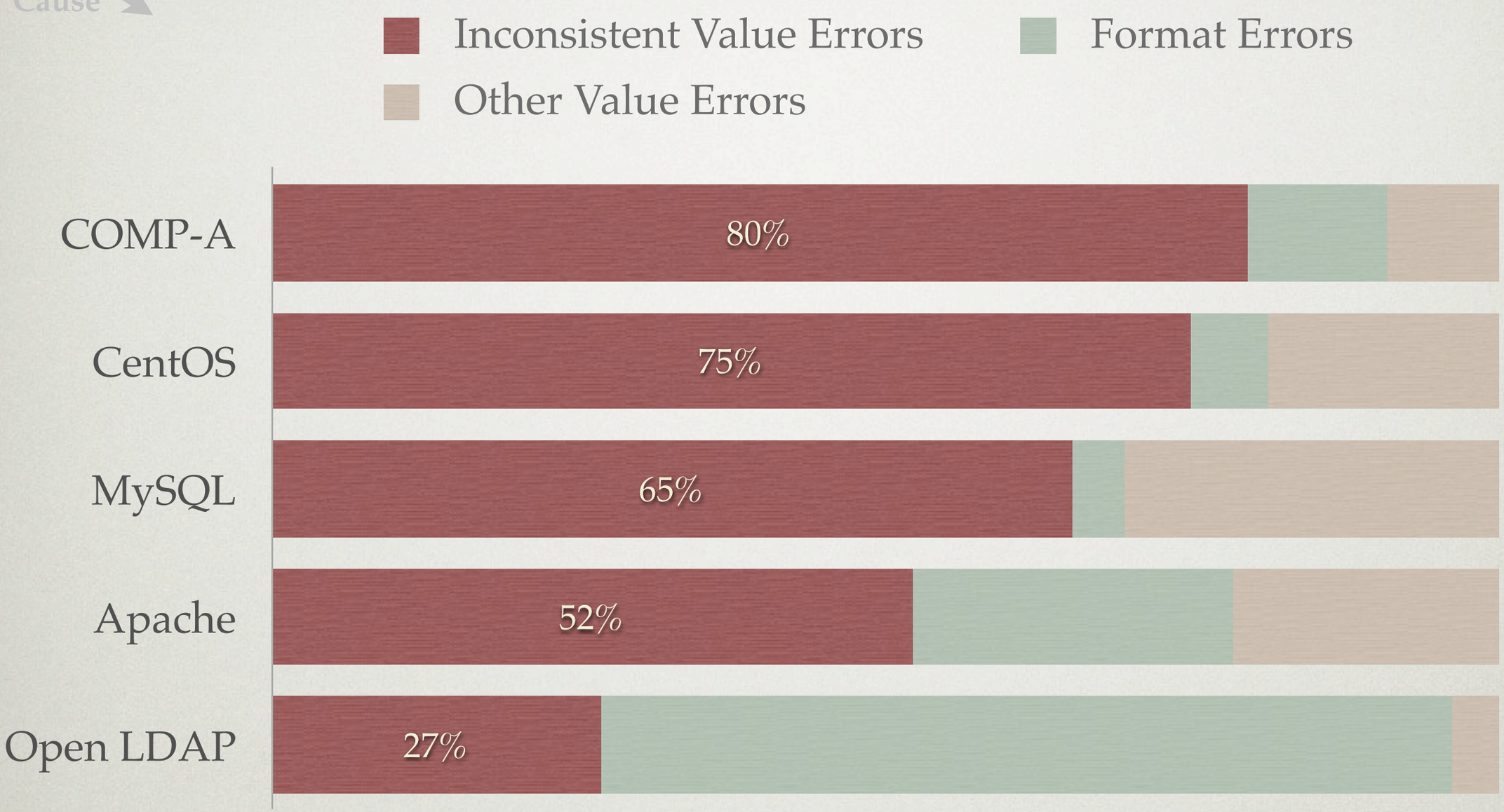
Value Errors

Format Errors

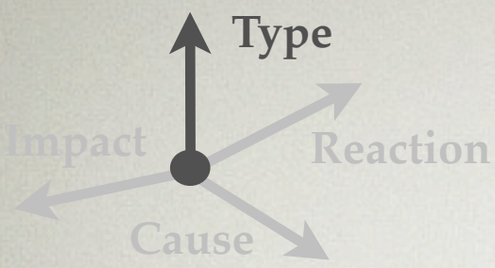
Inconsistent Values Errors

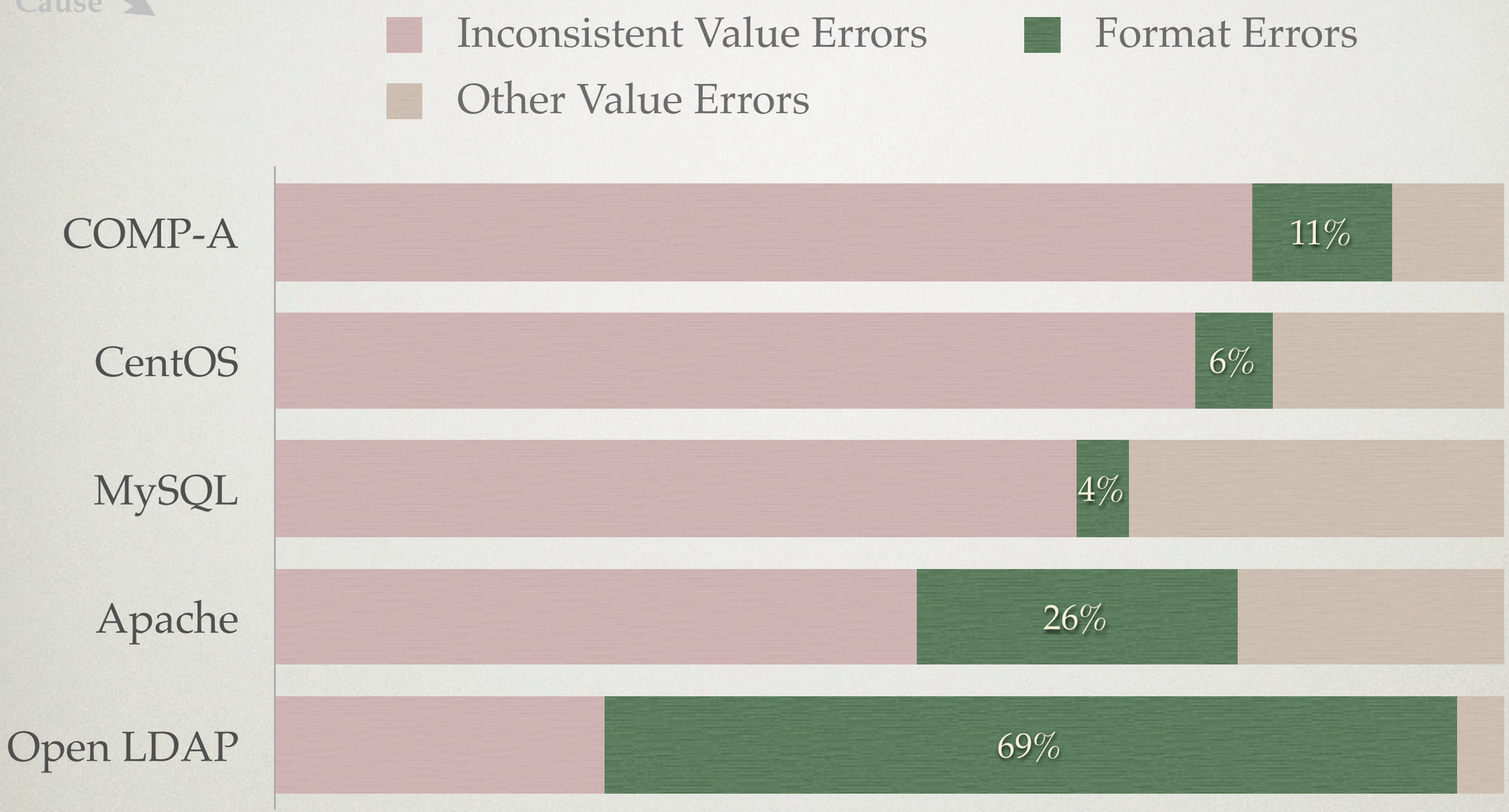
Other Value Errors



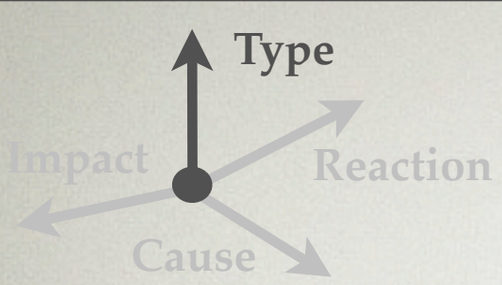


Inconsistent values dominate illegal parameter errors for most systems





All systems have format errors,
in particular, Open LDAP has 69%



Inconsistent Value Errors

Across Multiple Systems!

MySQL

```
log_output="Table"
...
log=query.log
```

Not consistent. They should be:

```
log_output="Table"
```

or

```
log_output="File"
log=query.log
```

PHP + MySQL

PHP configuration:

```
mysql.max_persistent = 400
```

MySQL configuration:

```
max_connections = 300
```

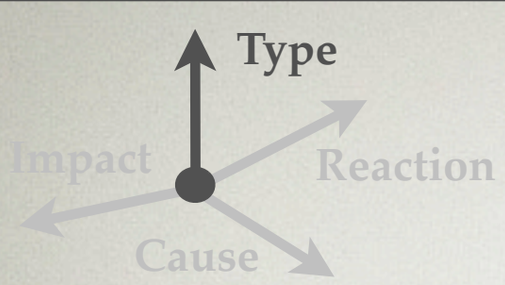
The value in PHP should not be bigger than the value in MySQL

- Value consistency constraints are error-prone; they account for most illegal parameter errors

- Consistency constraints could be across multiple systems, which is more difficult for users to follow

```
log_output="File"
log=query.log
```

not be bigger than the value in MySQL



Format Errors

COMP-A

```
InitiatorName: iqn_DEV_domain
```

Lower-case only value

Error

OpenLDAP

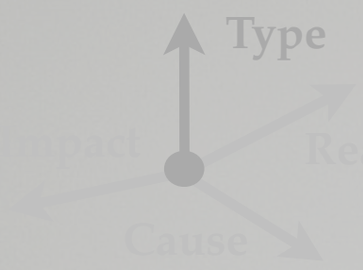
```
include schema/ppolicy.schema
.......
overlay ppolicy
```

Missing

Apache

```
extension = mysql.so
.......
extension = recode.so
```

`recode.so` must be put before `mysql.so`

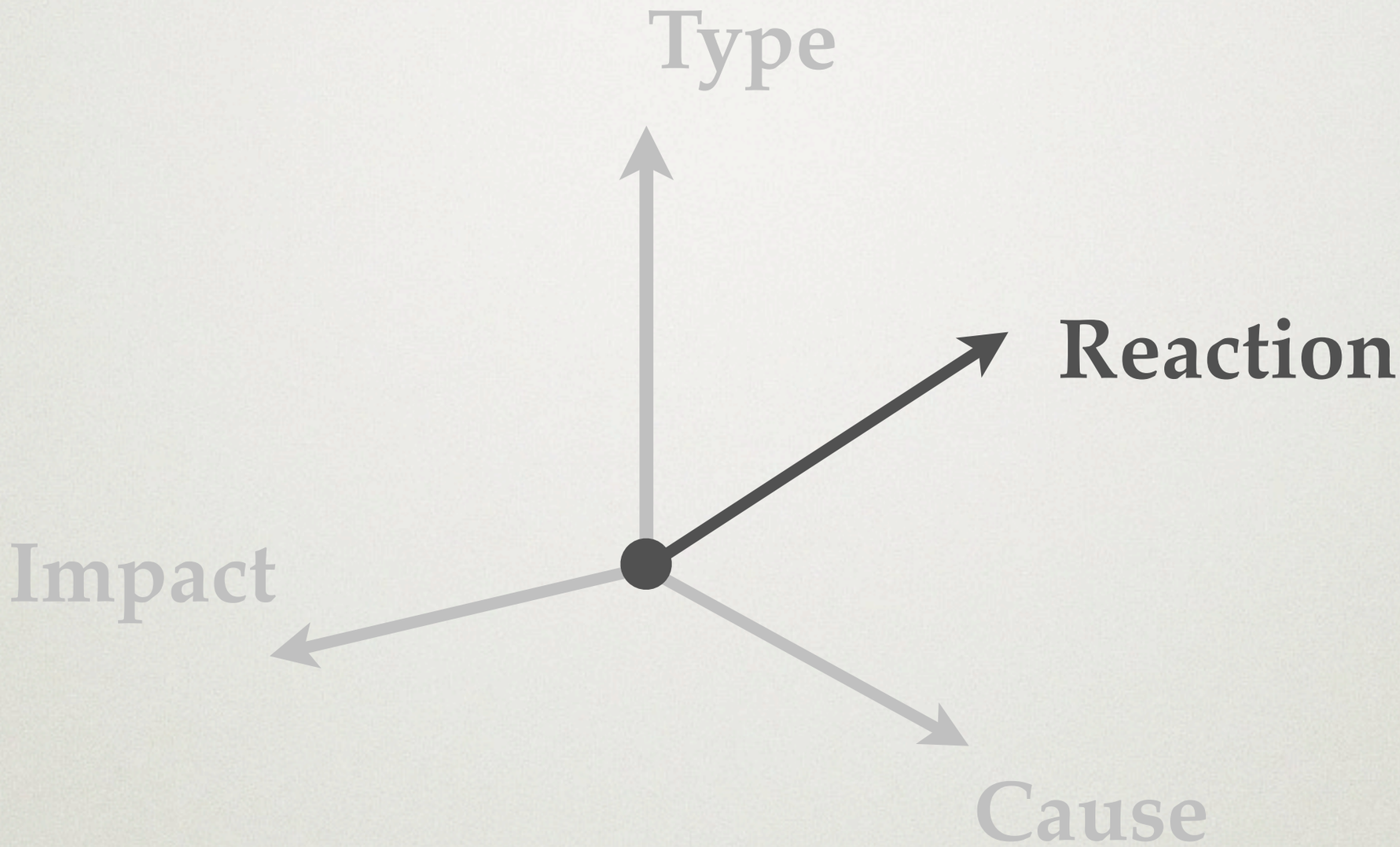


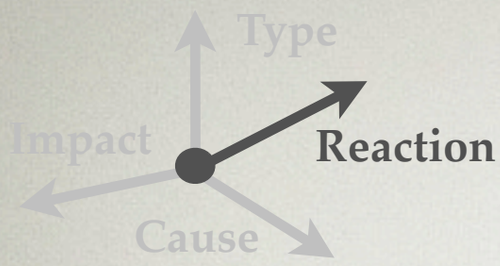
- Format constraints are difficult to follow, especially non-intuitive ones, e.g., upper-case vs. lower-case or ordering

- Format errors are relatively easier to detect compared to value errors

Missing

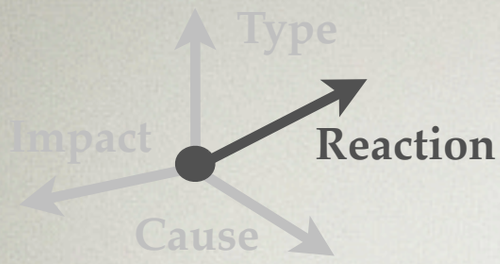
`recode.so` must be put before `mysql.so`


Type
Reaction
Impact
Cause



System Reaction

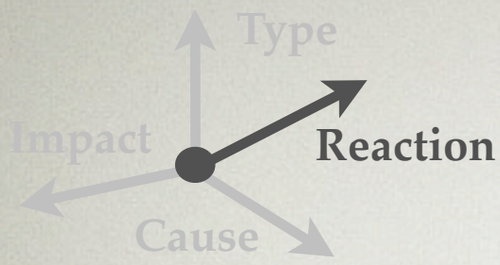
- Pinpoint Reaction
- Indeterminate Reaction
- Quiet Failure



Good Reaction #1

- **Symptom**: the user cannot create new directories in "`/vol/vol1/data/`"
- **Reaction**: the system prints this message:

```
[COMP-A – dir.size.max:warning]:
Directory /vol/vol1/data/ reached
the maxdirsize Limit. Reduce the number
of files or use the vol options command
to increase this limit.
```



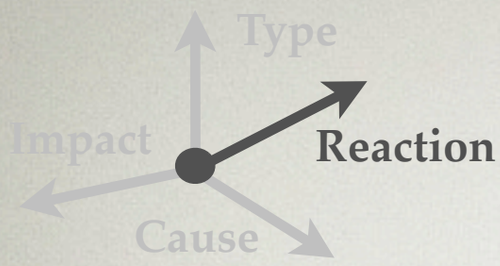
Good Reaction #2

MySQL

```
log_output=”Table”
...
log=query.log
```

Patch:

```
+if (opt_logname
+ && !(log_output_options & LOG_FILE)
+ && !(log_output_options & LOG_NONE))
+  sql_print_warning("Although a path was specified
+  for the --log option, log tables are used. To enable
+  logging to files use the --log-output option.");
```



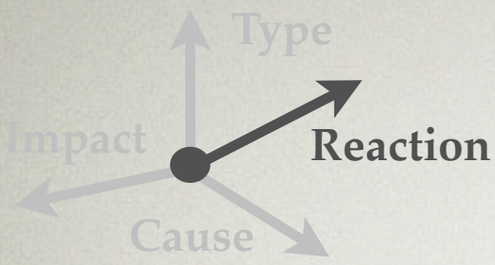
Good Reaction #2

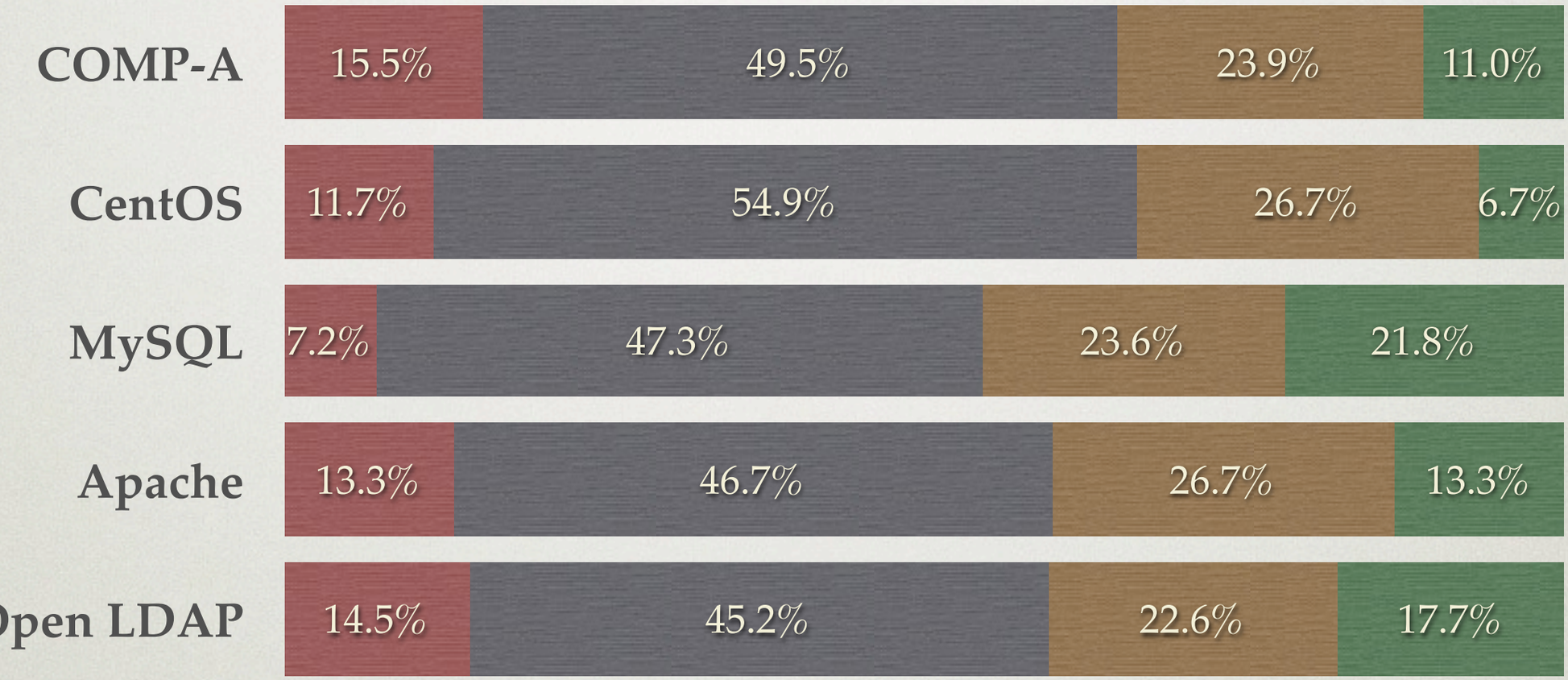
MySQL

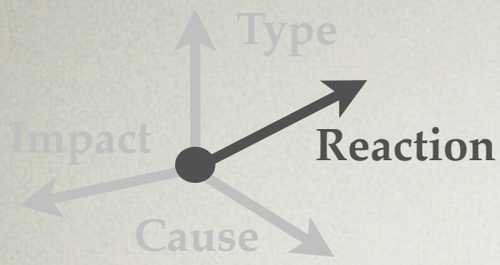
```
log_output=”Table”
...
log=query.log
```

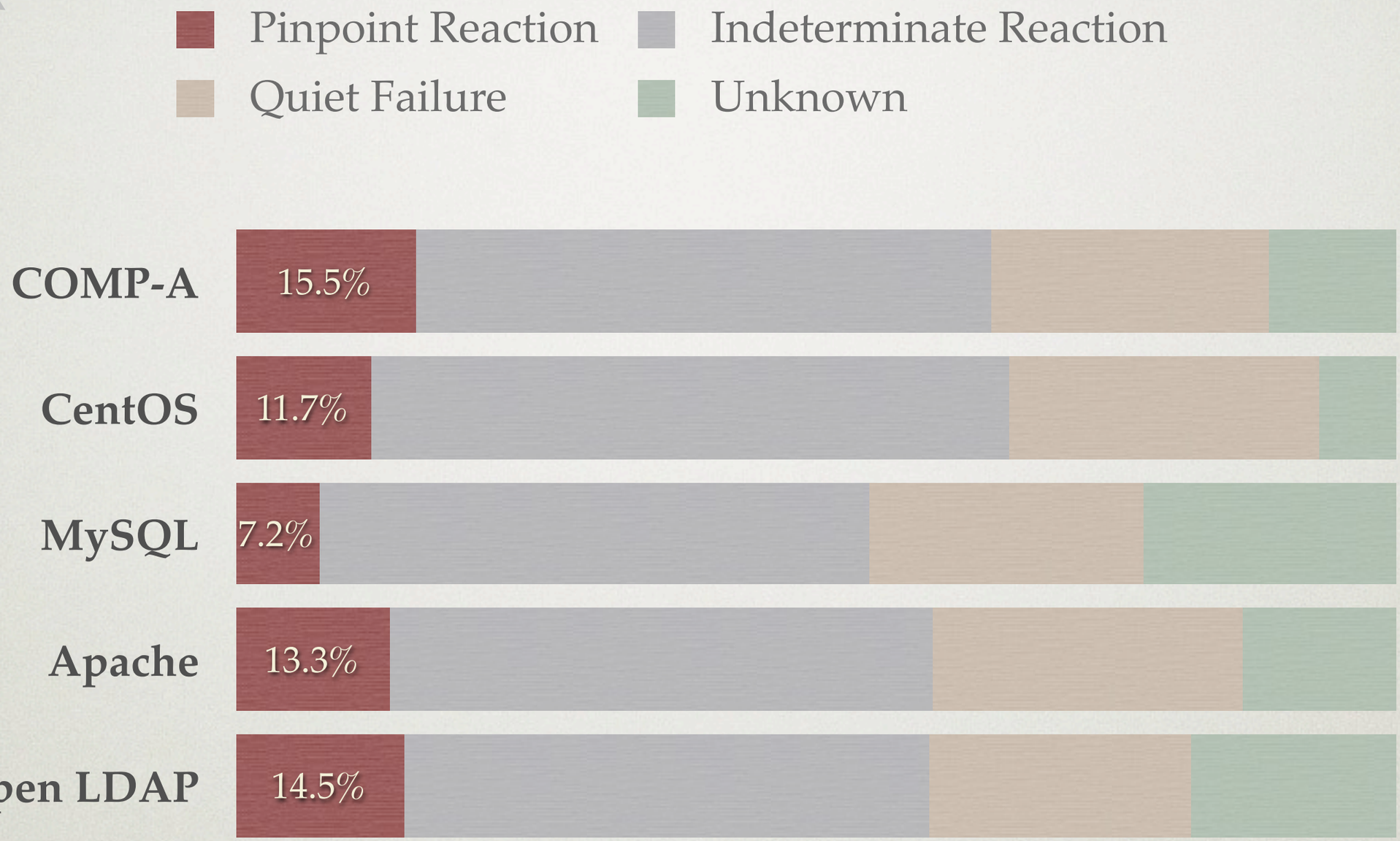
Patch:

```
+if (opt_logname
+ && !(log_output_options & LOG_FILE)
+ && !(log_output_options & LOG_NONE))
+  sql_print_warning("Although a path was specified
+  for the --log option, log tables are used. To enable
+  logging to files use the --log-output option.");
```

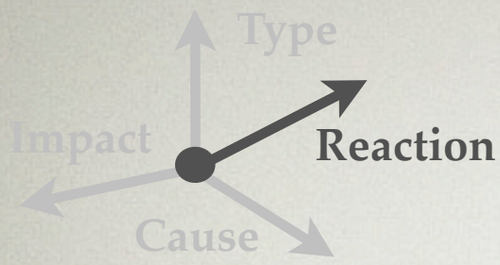

Type
Impact
Reaction
Cause

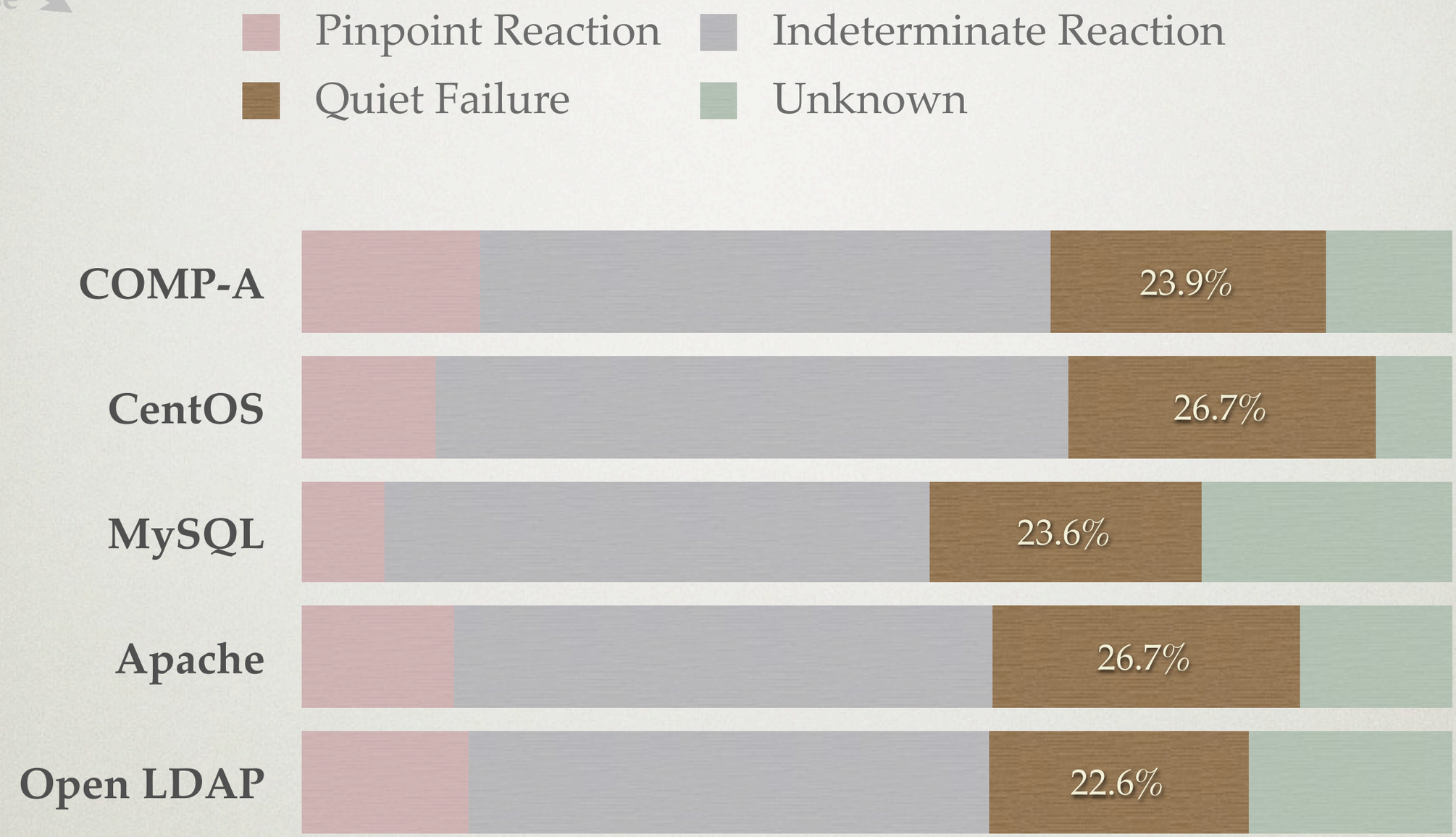

Pinpoint Reaction
Indeterminate Reaction
Quiet Failure
Unknown
COMP-A
15.5%
49.5%
23.9%
11.0%
CentOS
11.7%
54.9%
26.7%
6.7%
MySQL
7.2%
47.3%
23.6%
21.8%
Apache
13.3%
46.7%
26.7%
13.3%
Open LDAP
14.5%
45.2%
22.6%
17.7%



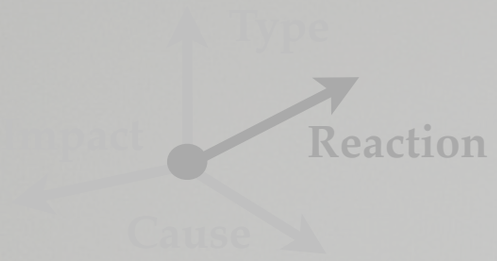


Today's systems do not react to configuration errors in a user-friendly way





Big portion of quiet failures makes diagnosis difficult

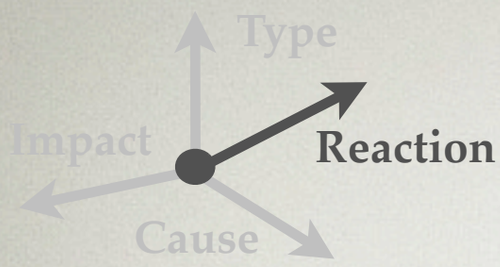


Pinpoint Reaction Indeterminate Reaction

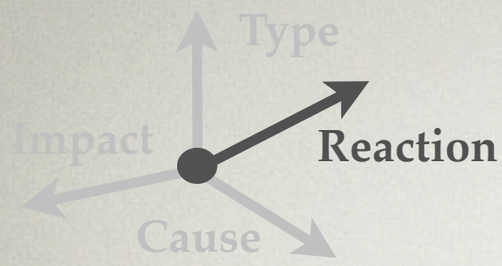
- Systems should avoid “bug-like” symptoms when configuration errors happen, such as quite failures, crash or hang

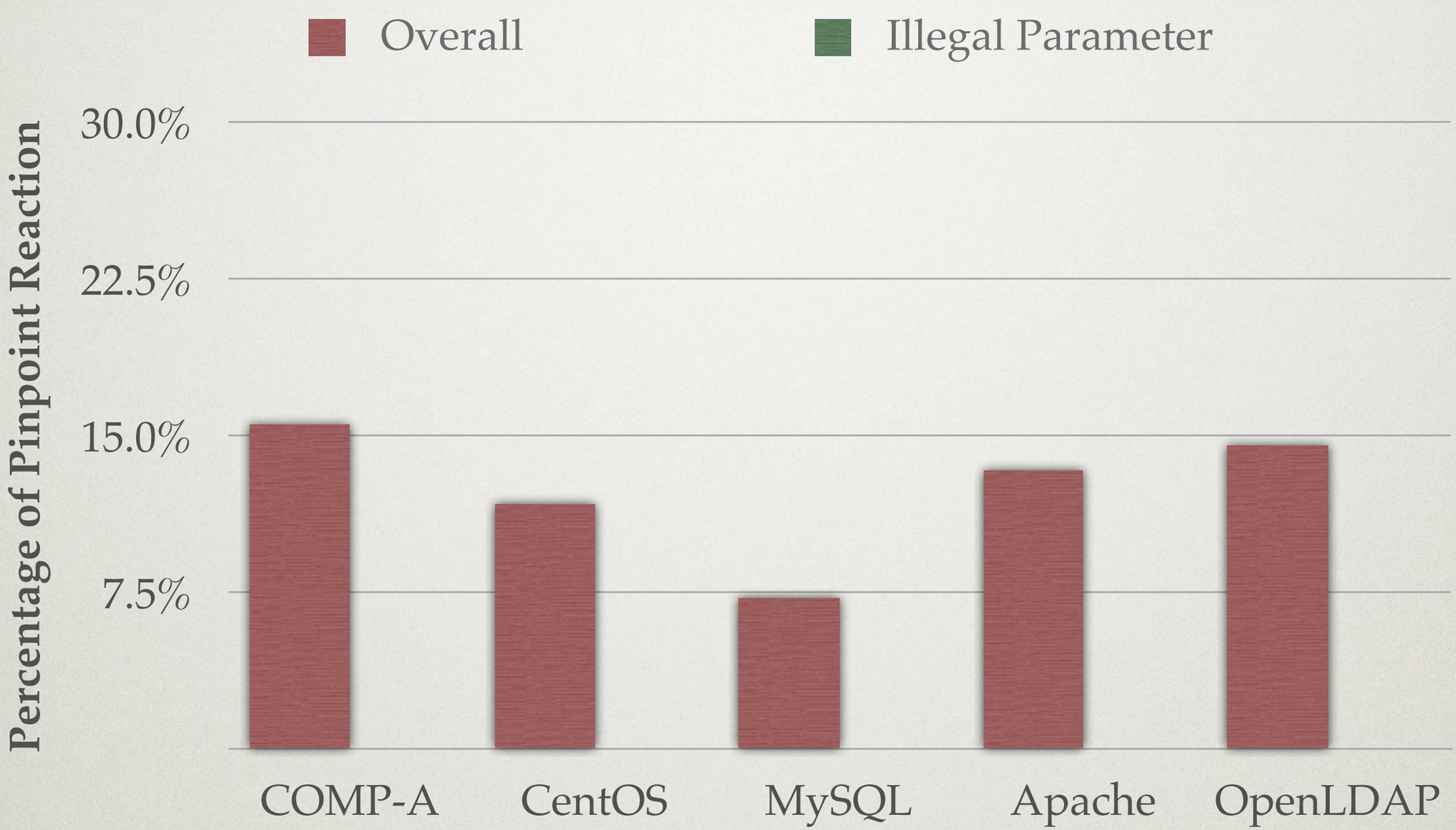
Open LDAP 22.6%

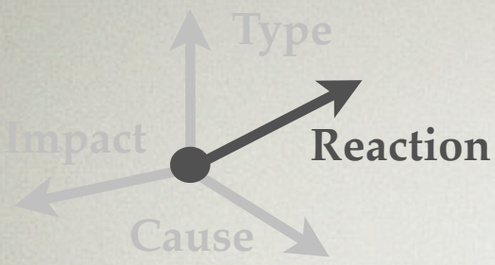
Big portion of quiet failures makes diagnosis difficult

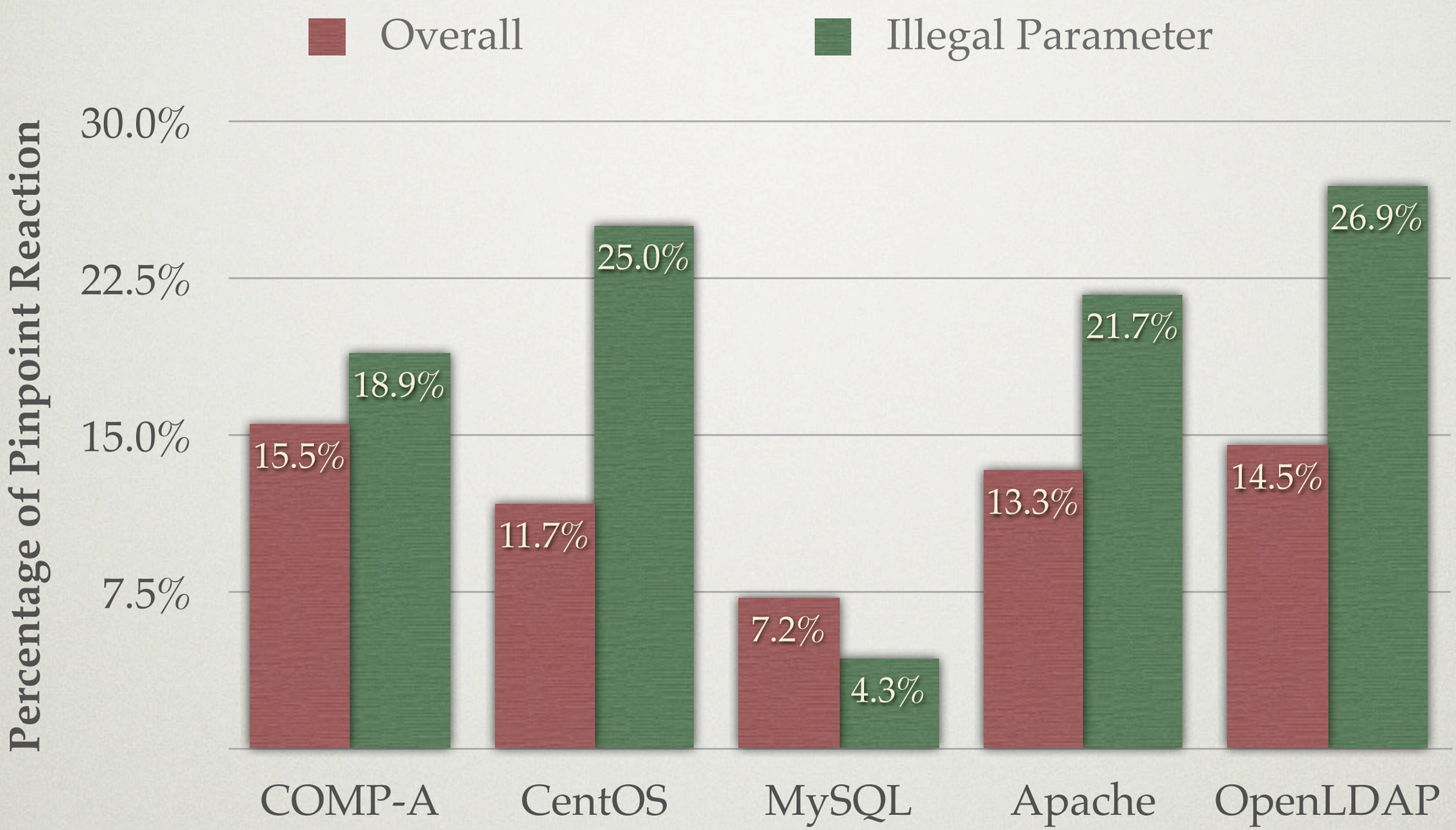


Do systems react better to errors with “illegal” parameters?

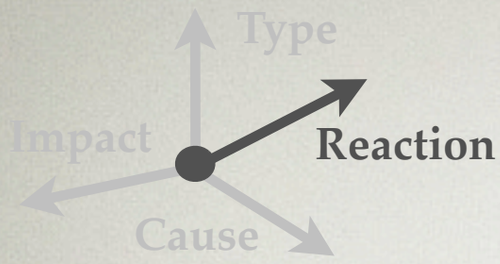

Type
Impact
Reaction
Cause


Overall
Illegal Parameter
Percentage of Pinpoint Reaction
30.0%
22.5%
15.0%
7.5%
COMP-A
CentOS
MySQL
Apache
OpenLDAP

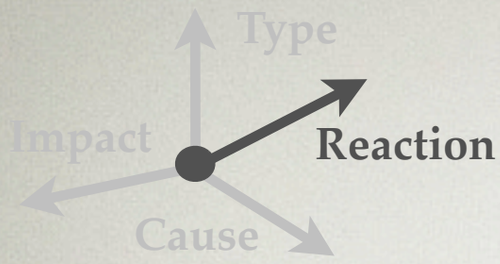




Illegal parameter errors are handled better, but not good enough

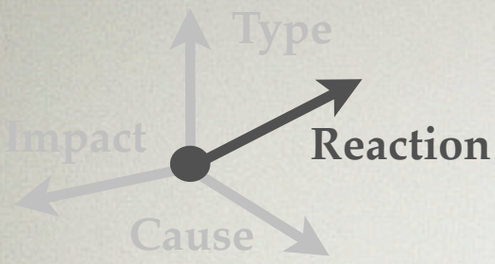


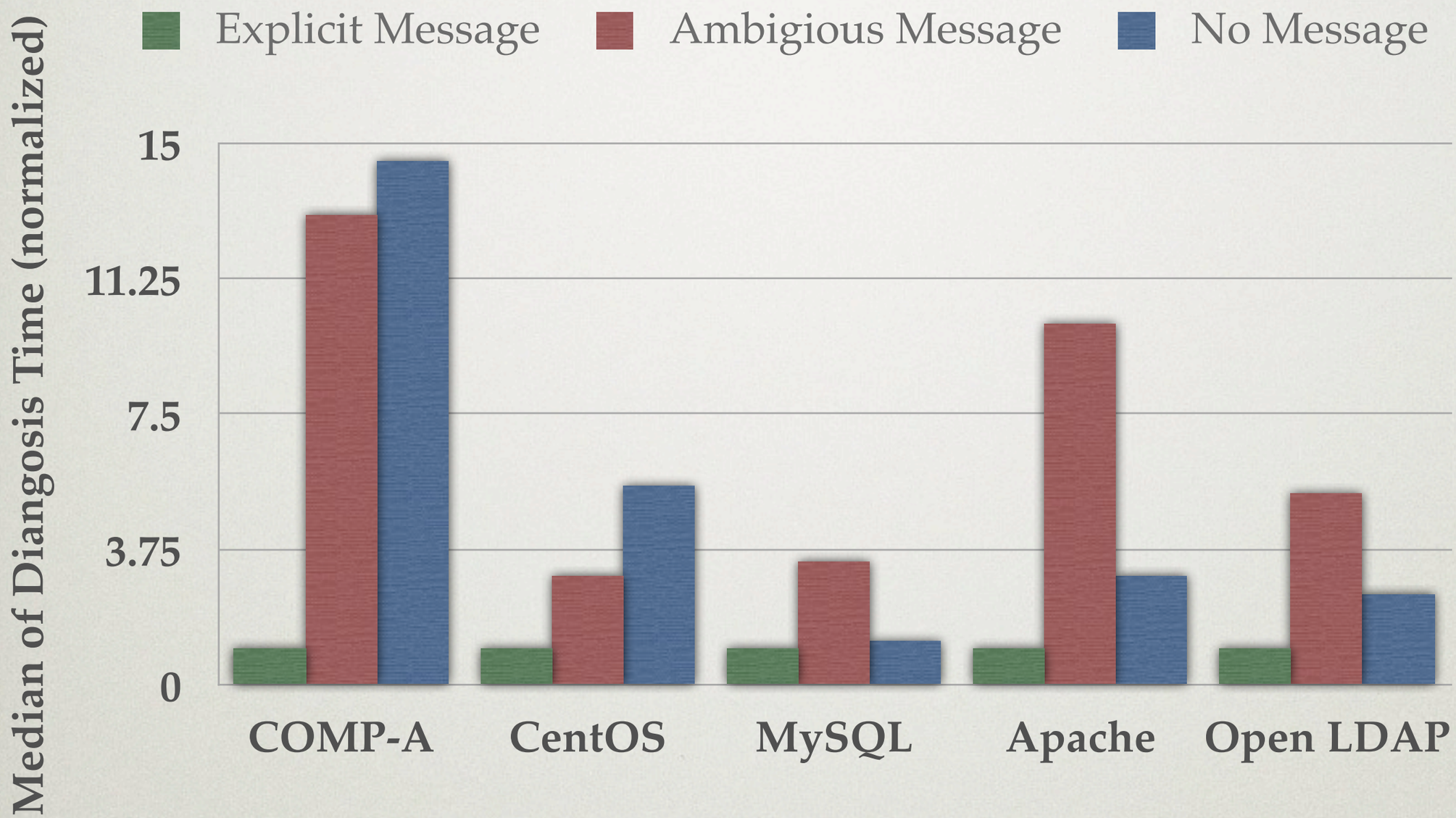
How does message quality affect diagnosis time?



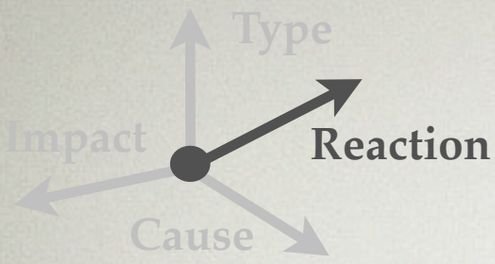
Message Quality

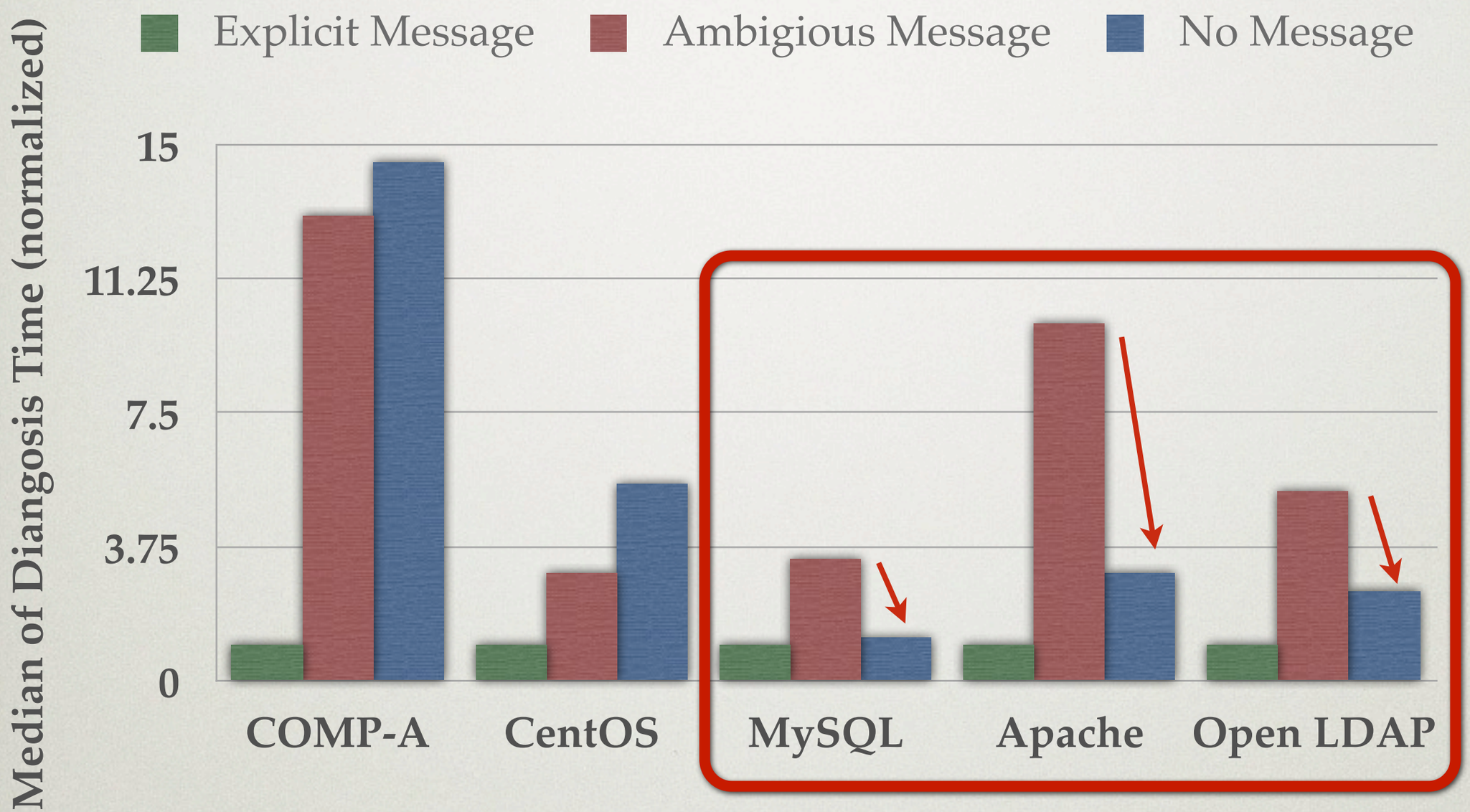
- Explicit Message
- Ambiguous Message
- No Message



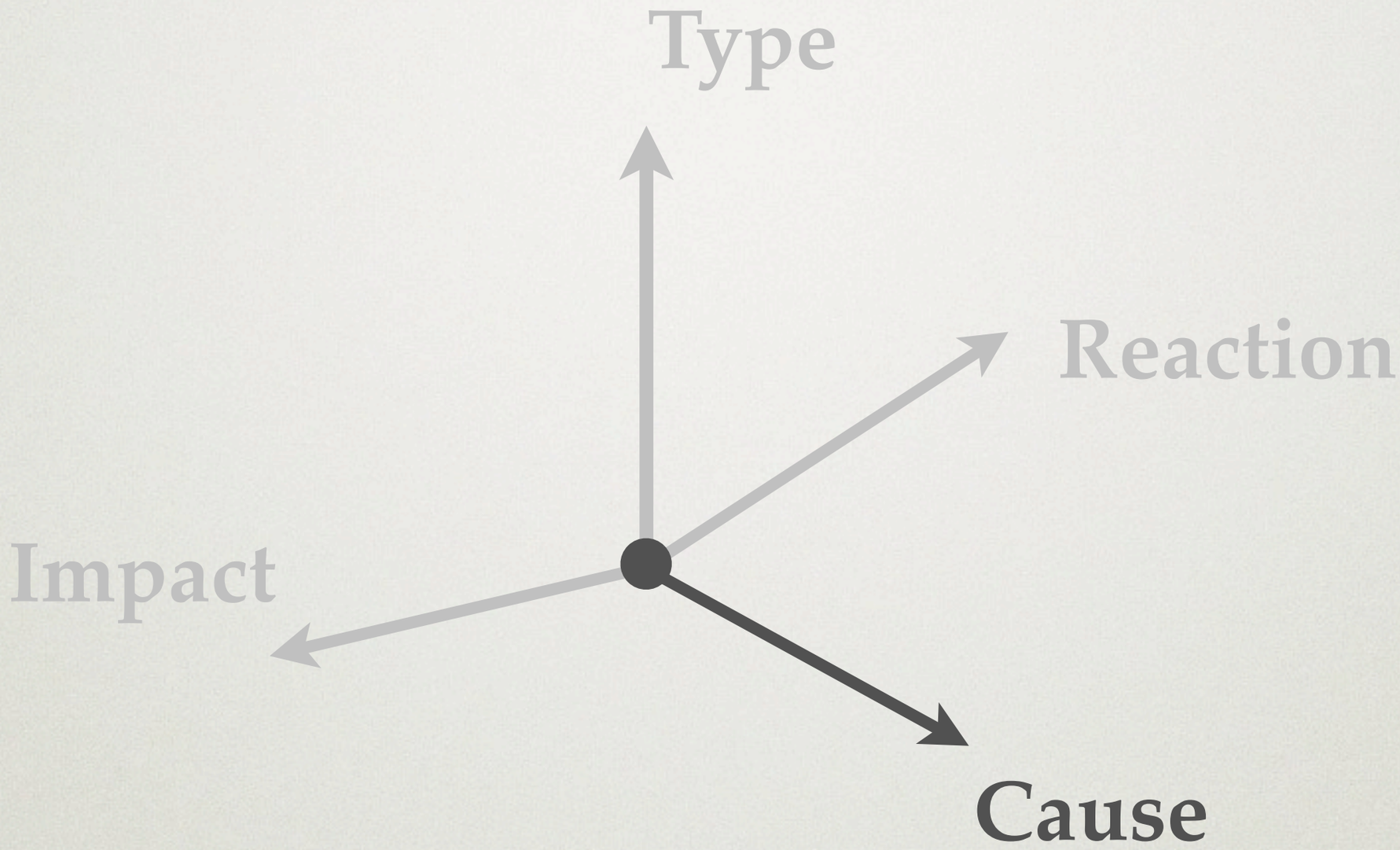


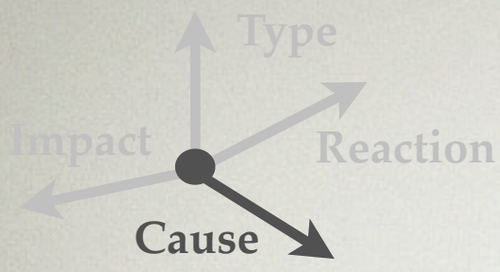
Explicit messages significantly reduce diagnosis time





Messages are harmful if they are misleading

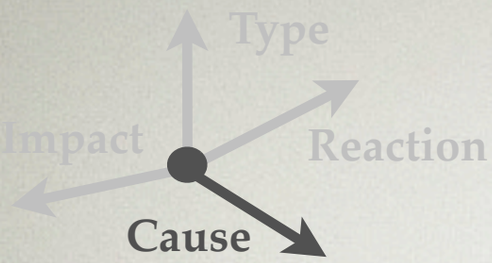

Type
Reaction
Impact
Cause

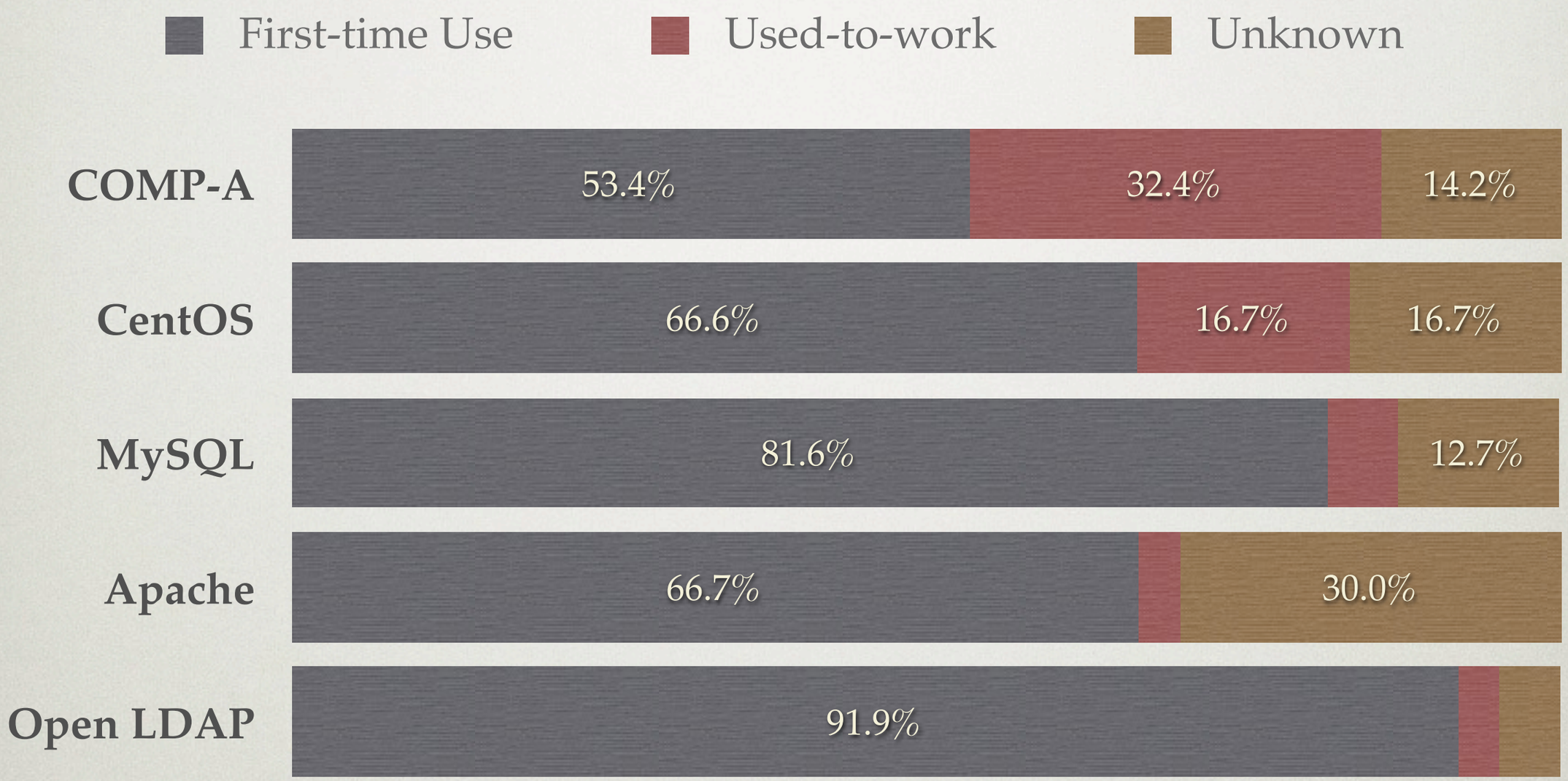


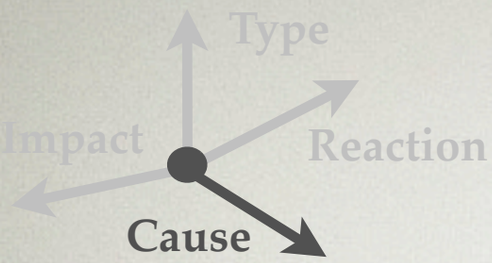
When does a configuration error happen?

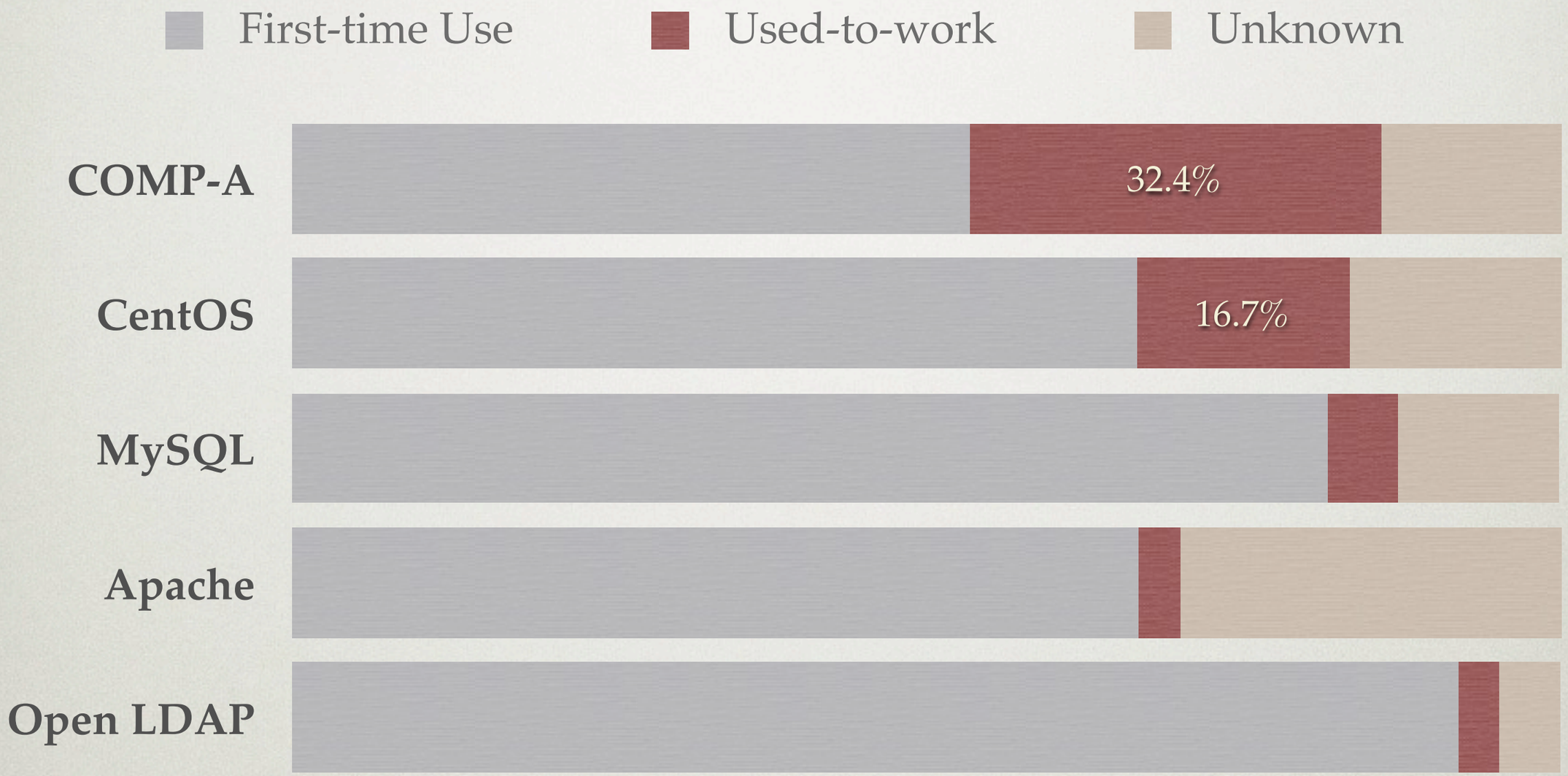
First-time Use

Used-to-work

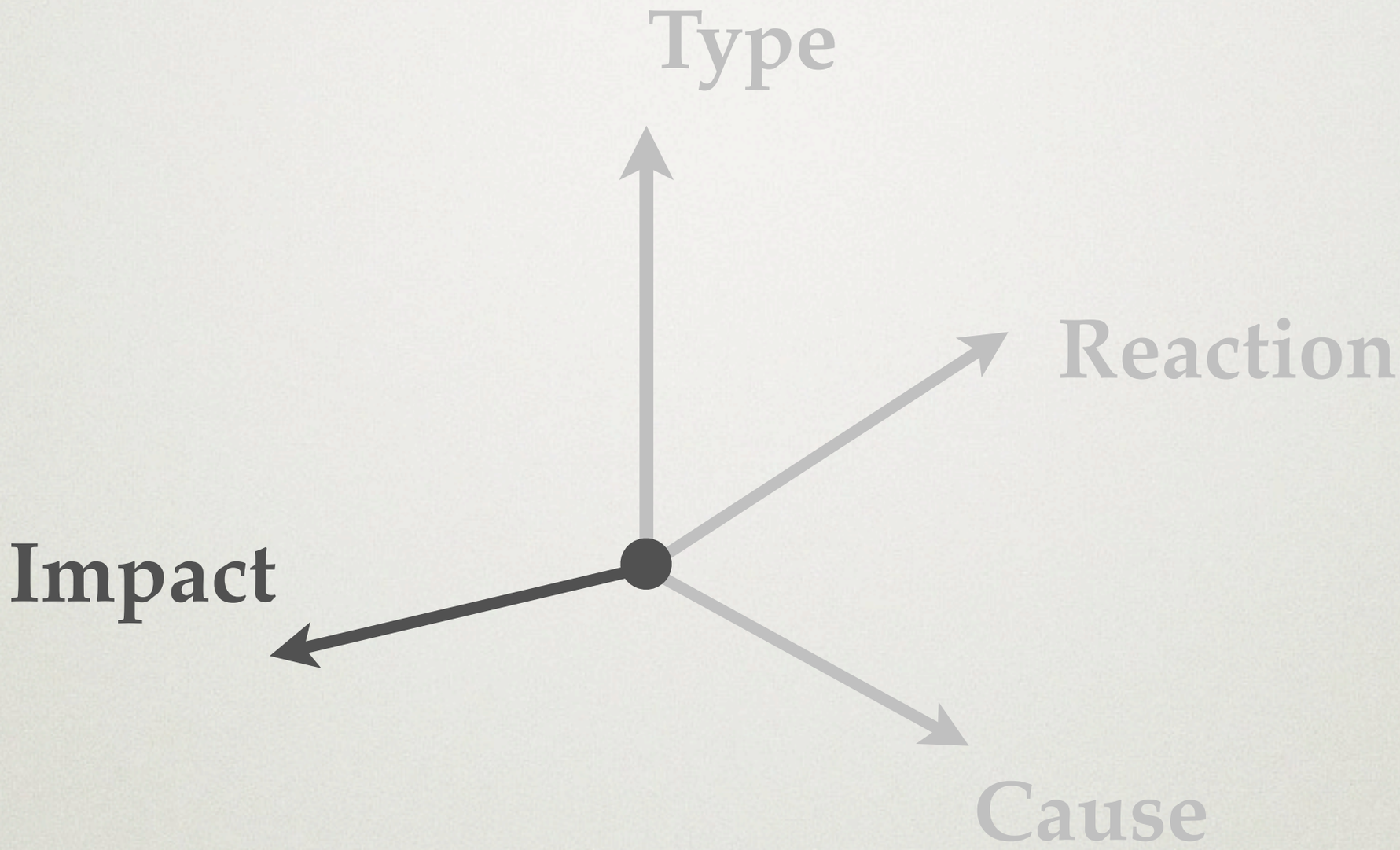

Type
Impact
Reaction
Cause

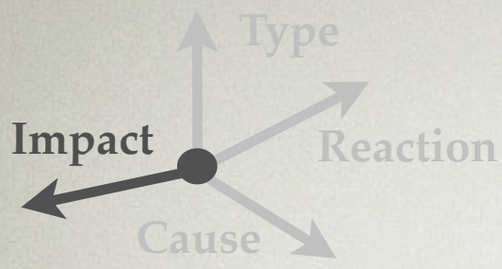

First-time Use
Used-to-work
Unknown
COMP-A
53.4%
32.4%
14.2%
CentOS
66.6%
16.7%
16.7%
MySQL
81.6%
12.7%
Apache
66.7%
30.0%
Open LDAP
91.9%





Complex systems are more likely to have configuration errors in the middle of lifetime


Type
Reaction
Impact
Cause

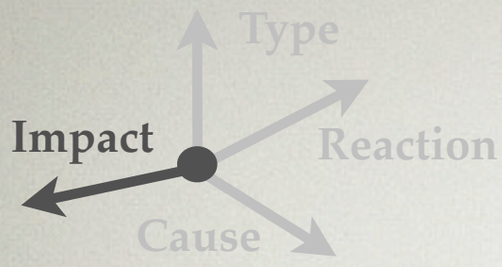


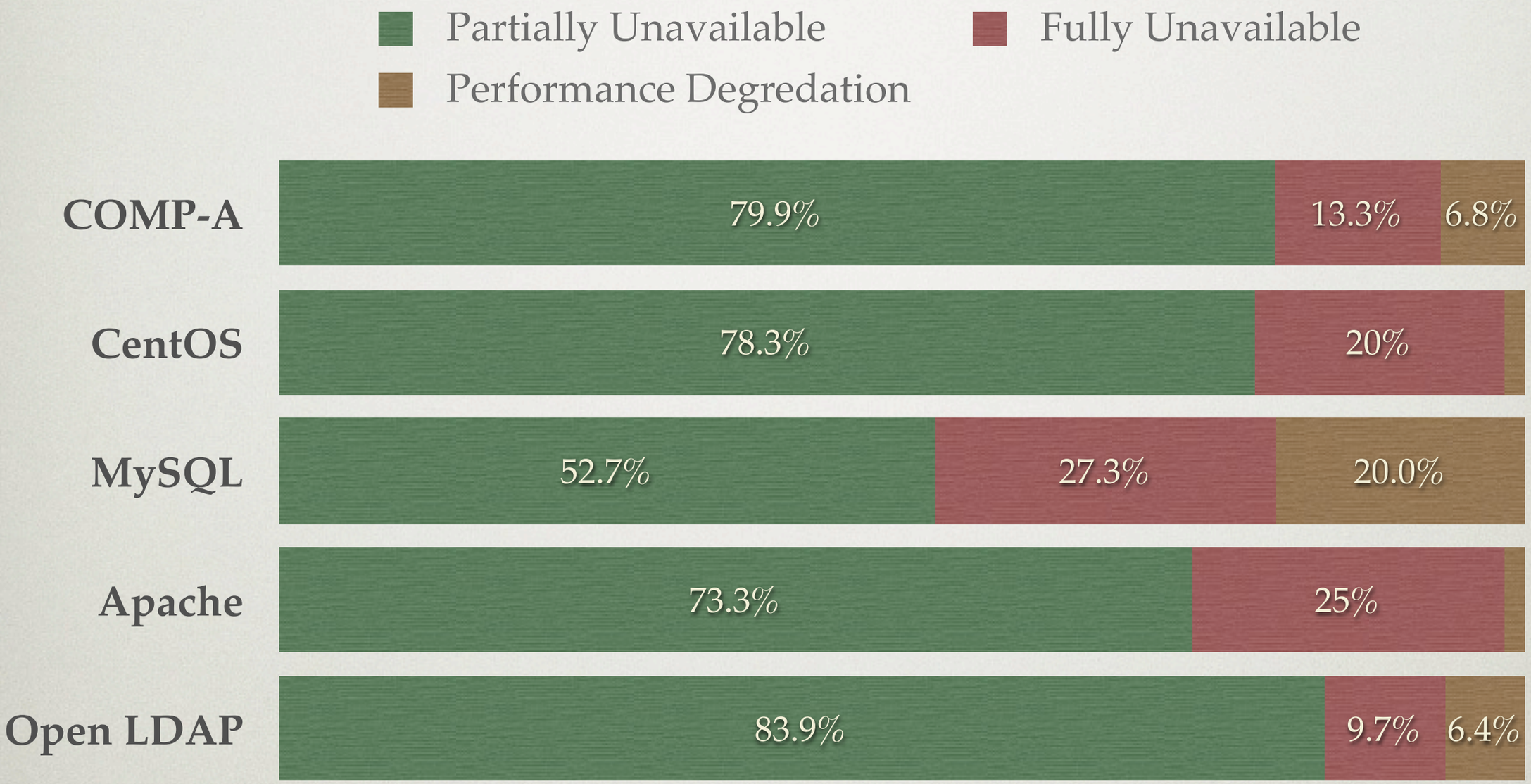
How do configuration errors affect system availability?

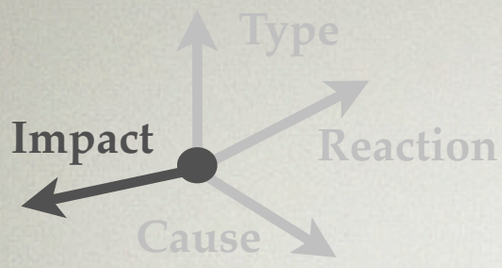
Partially Unavailable

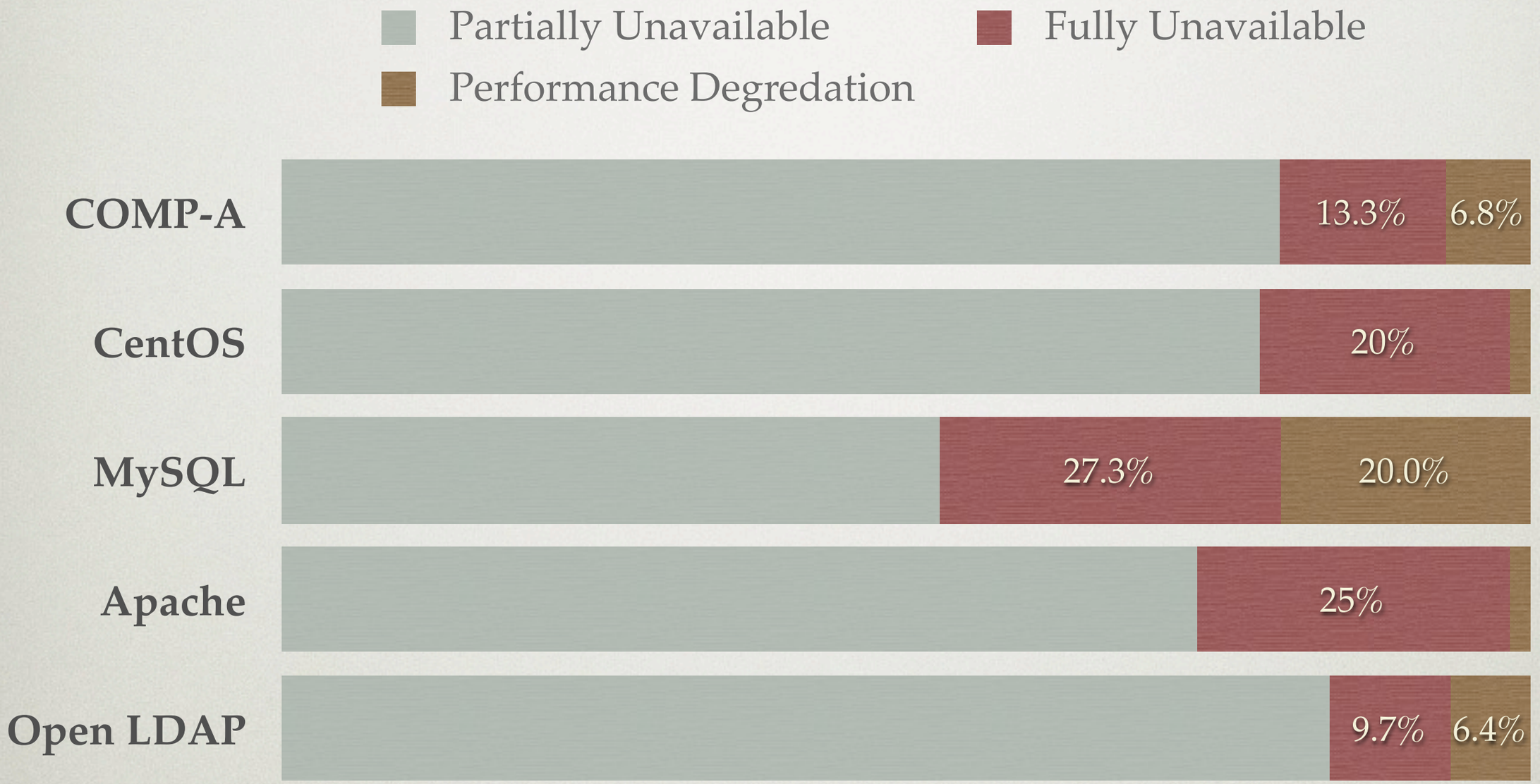
Fully Unavailable

Performance Degradation

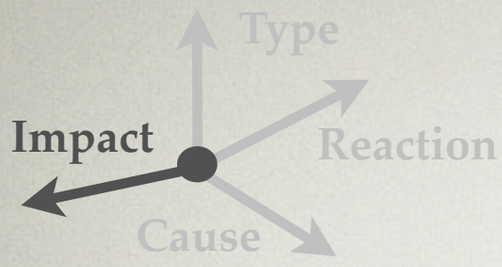

Type
Impact
Reaction
Cause

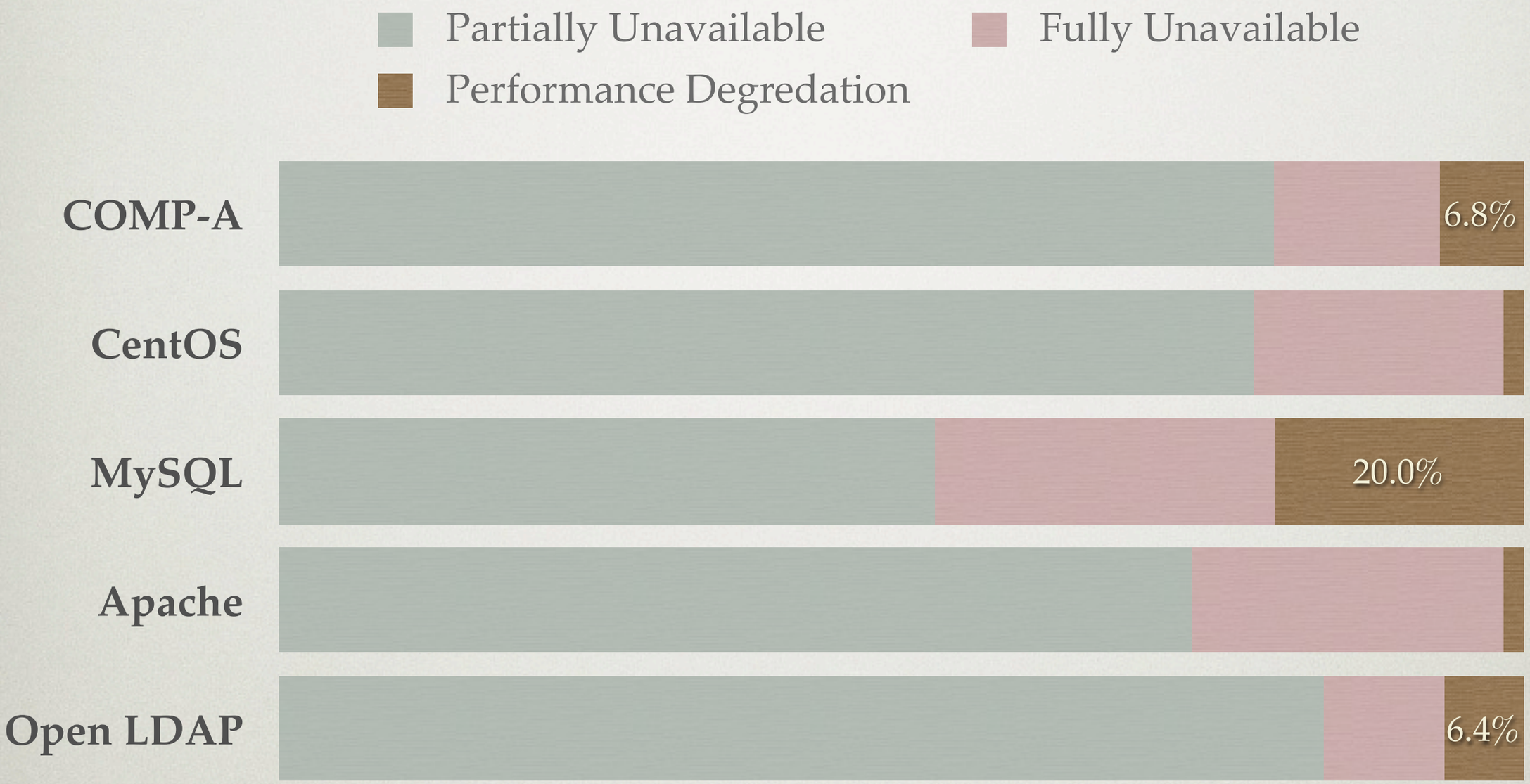

Partially Unavailable
Fully Unavailable
Performance Degredation
COMP-A
79.9%
13.3%
6.8%
CentOS
78.3%
20%
MySQL
52.7%
27.3%
20.0%
Apache
73.3%
25%
Open LDAP
83.9%
9.7%
6.4%



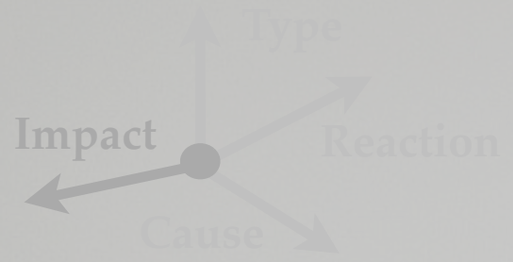


Configuration errors can cause system full unavailability and performance degradation





Performance configuration is especially difficult



- Performance parameters are more difficult to understand and set
- Diagnosing performance configuration issues are troublesome

Performance configuration is especially difficult

Other Characteristics

- Location and domain of parameter errors
- Number of involved/fixed parameters
- Complete categorization of illegal parameters and their distribution
- Complete analysis of causes of errors
- Details about compatibility errors and component errors
- Analysis across multiple directions
- More examples

Related Work

- **Prevention** (SmartFrog, Kardo, etc.)
- **Detection** (PeerPressure, Strider, etc.)
- **Diagnosis** (Chronus, AutoBash, ConfAid, etc.)
- **Tolerance** (AutoBash, Undo, etc.)
- **Validation** (Barricade, etc.)
- **Injection/testing** (ConfErr, etc.)

Summary

- **Think** from users' point of view
  - Users do not know the code
  - Keep things simple
    - Expose fewer "knobs"
    - Use intuitive and simple rules
  - Validate configuration proactively
  - React decently when configuration errors happen
  - Provide good feedback to users
- **Record** configuration errors
- **Learn** from mistakes

Thank you!