

# Identifying Crosscutting Concerns Using Historical Code Changes

Bram Adams  
Zhen Ming Jiang  
Ahmed E. Hassan

SAIL, Queen's University

<http://sailhome.cs.queensu.ca/~bram/>

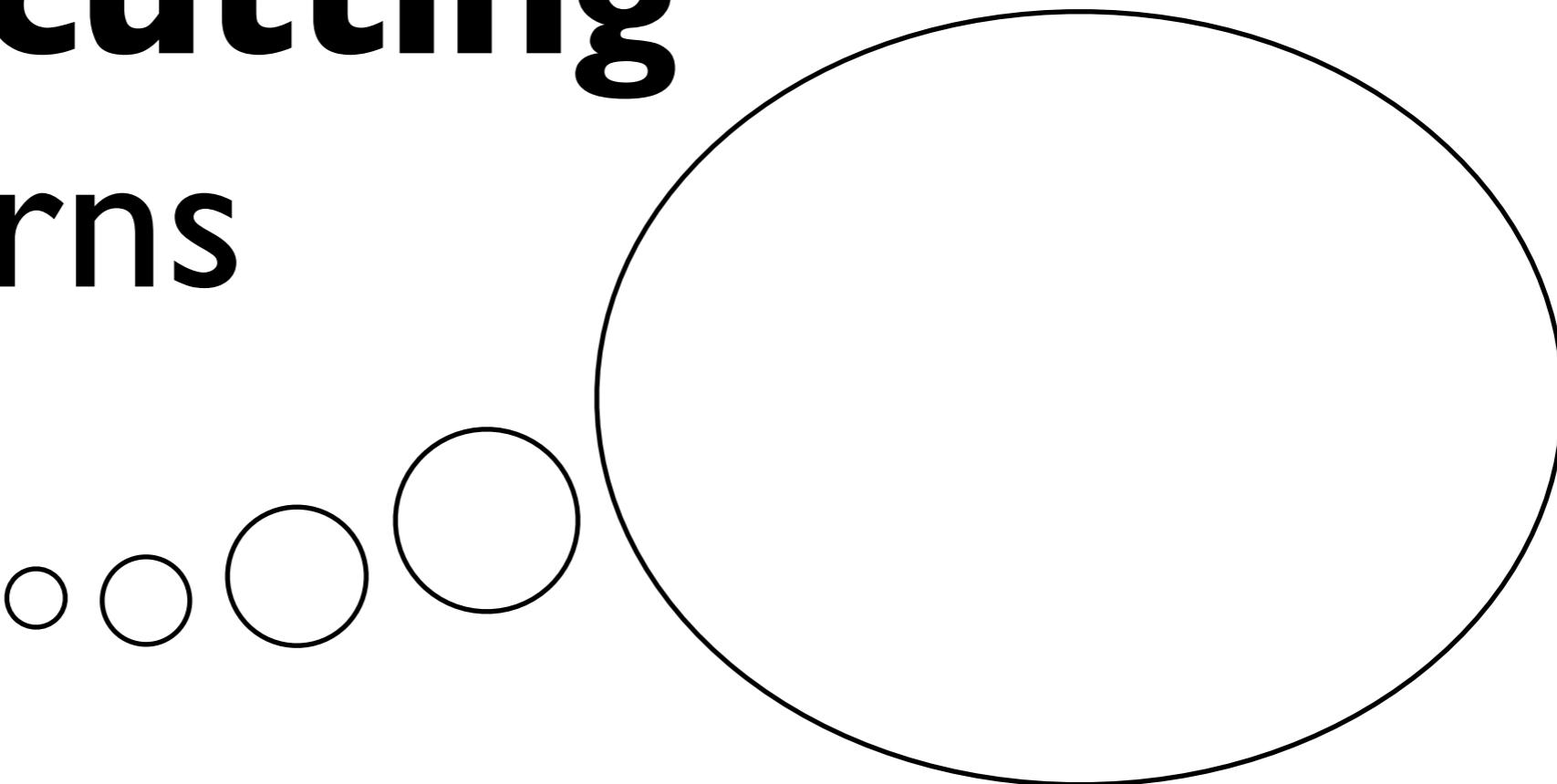




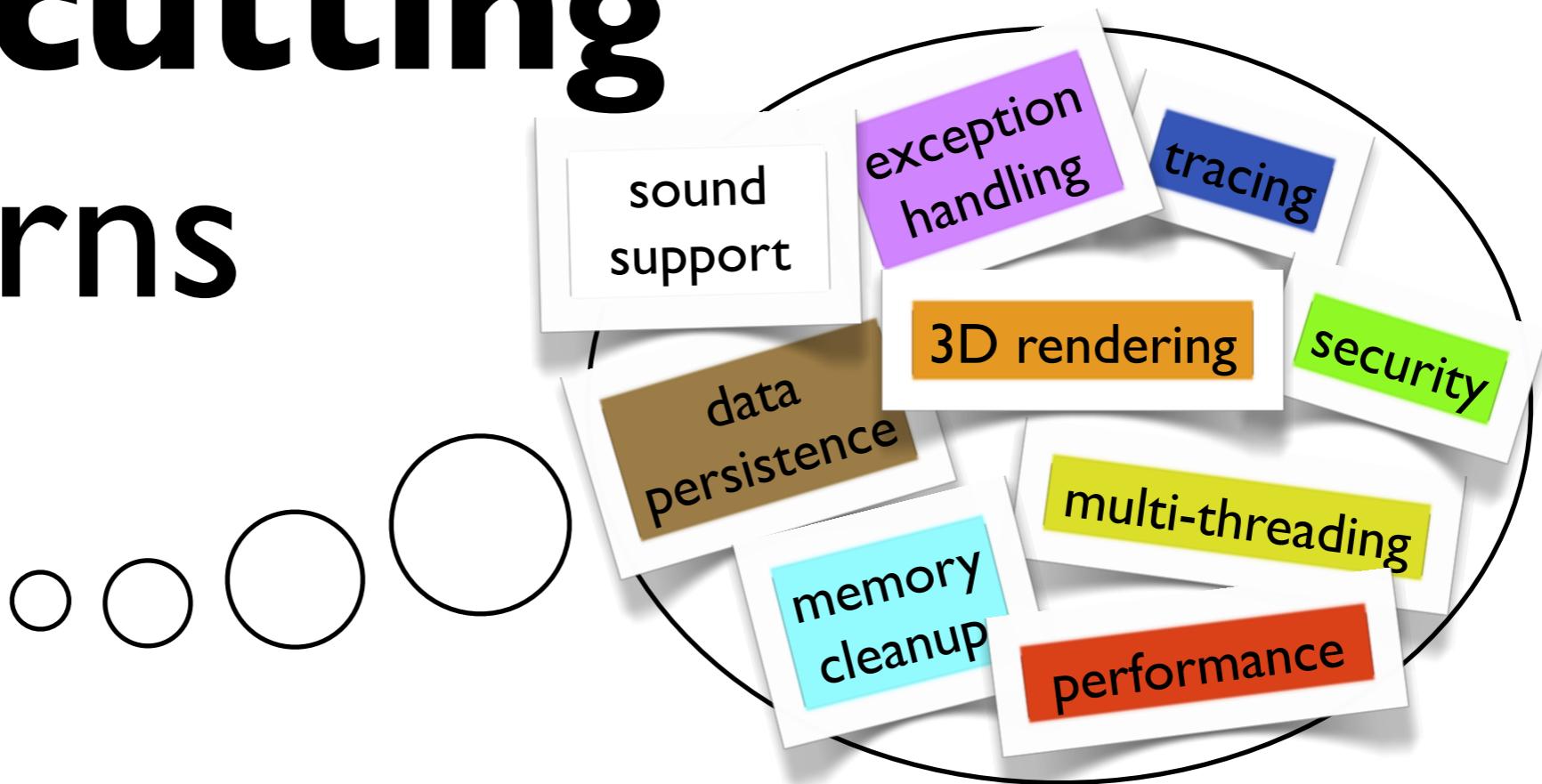
**What are crosscutting concerns?**

# **Crosscutting Concerns**

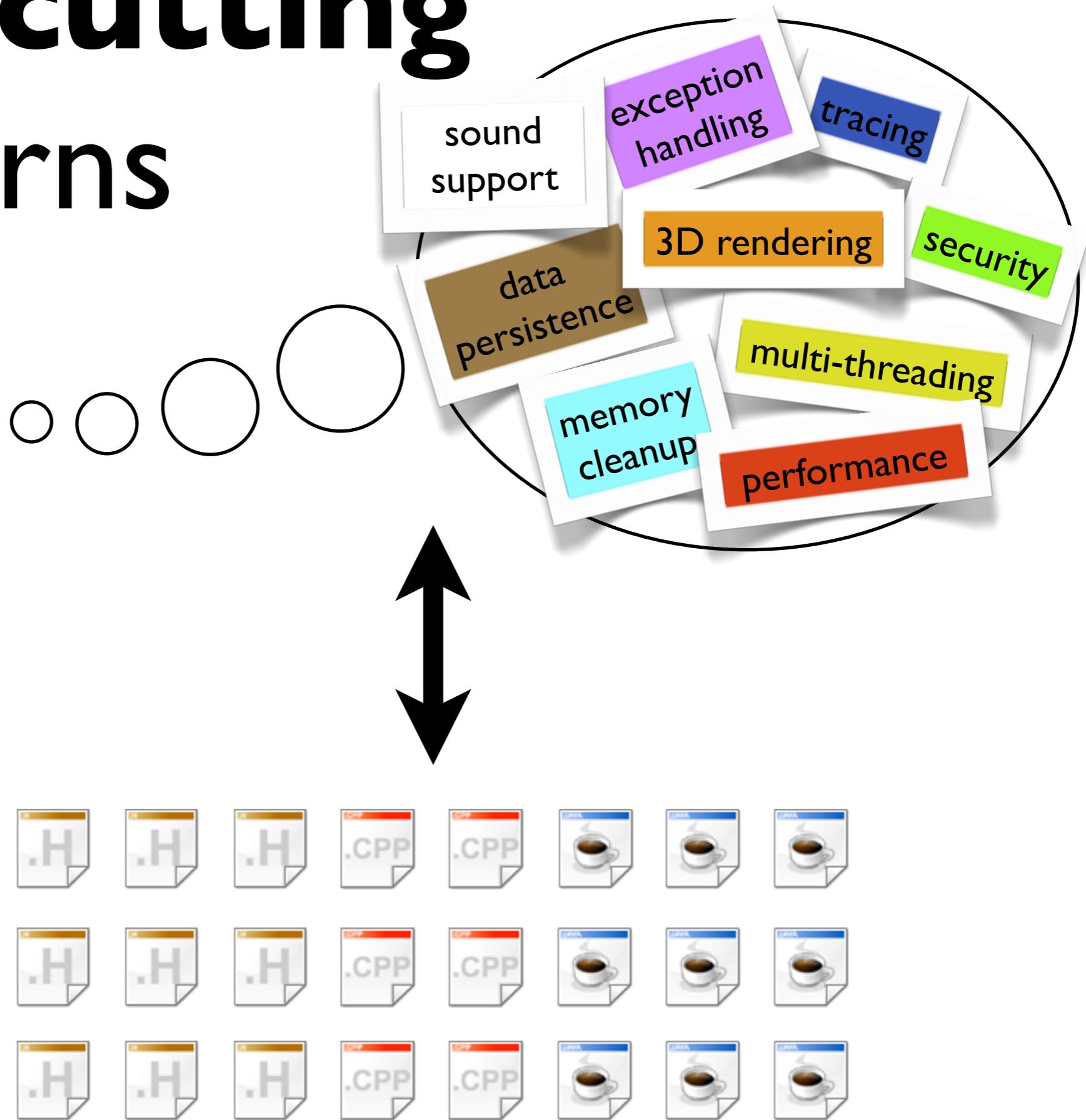
# Crosscutting Concerns



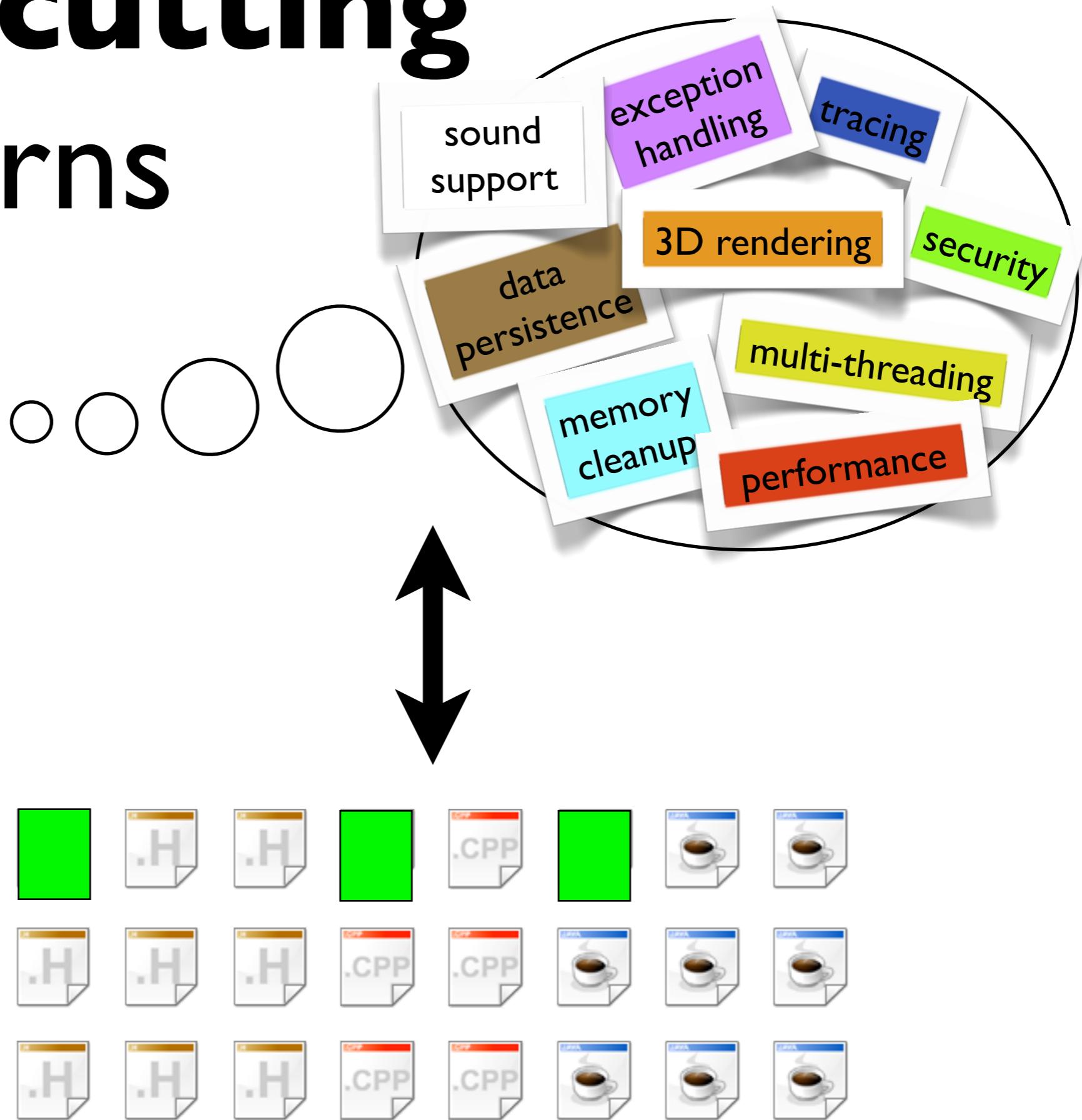
# Crosscutting Concerns



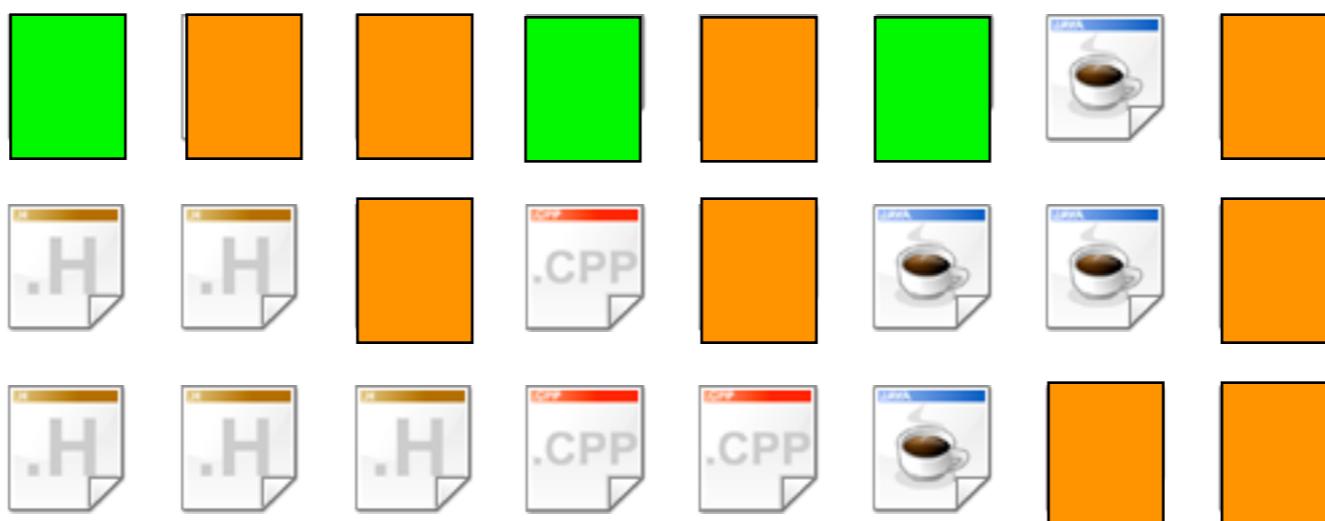
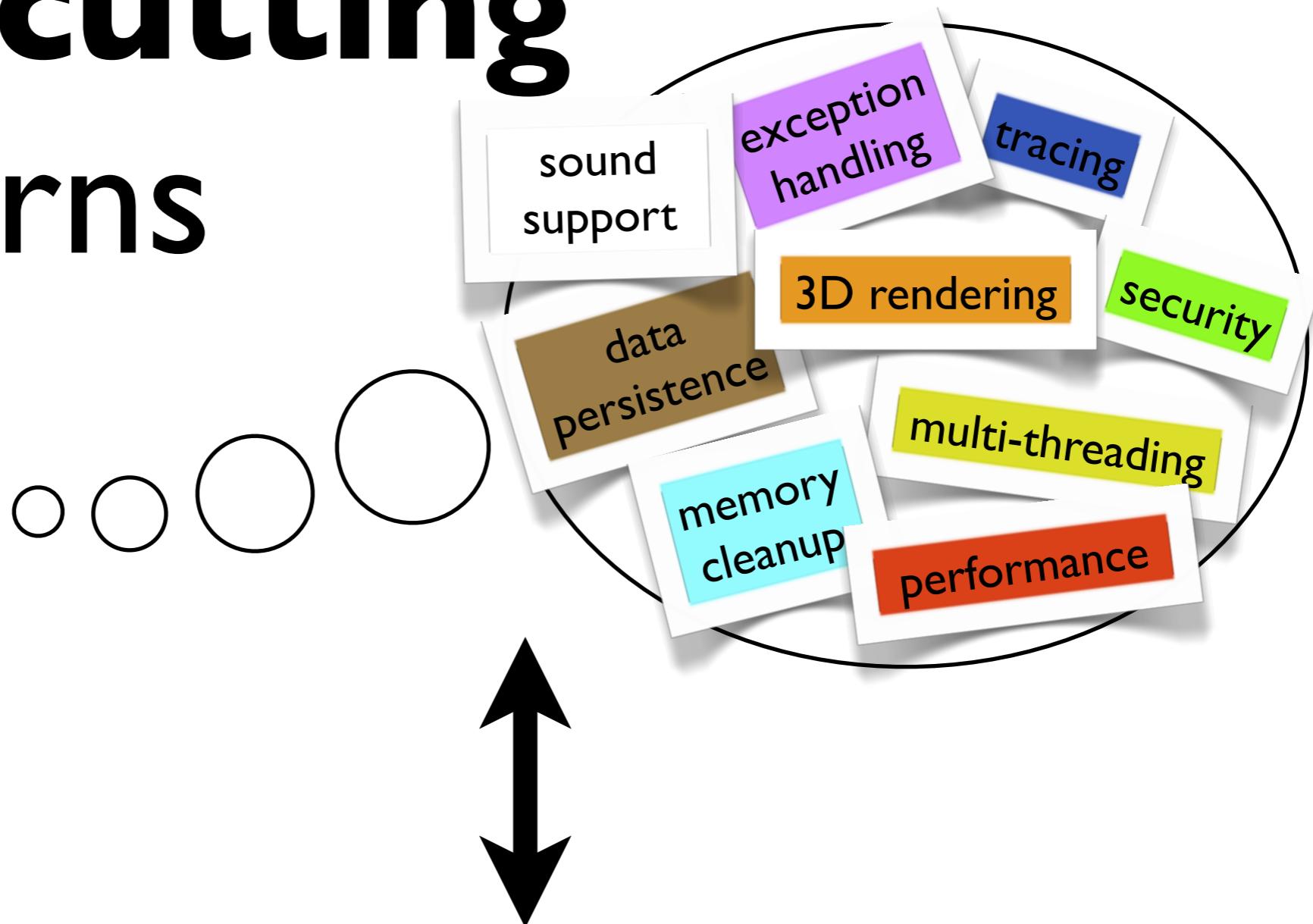
# Crosscutting Concerns



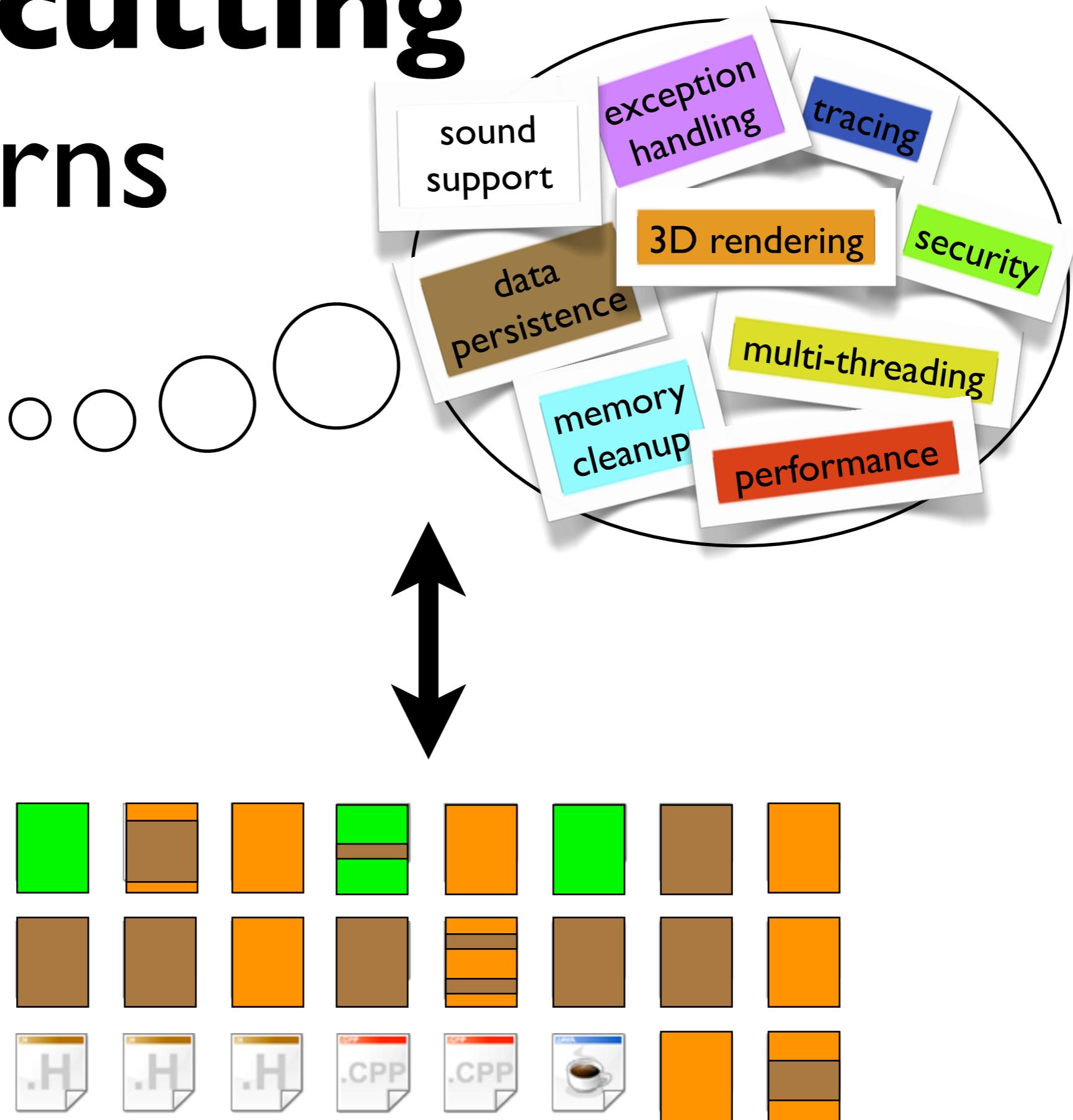
# Crosscutting Concerns



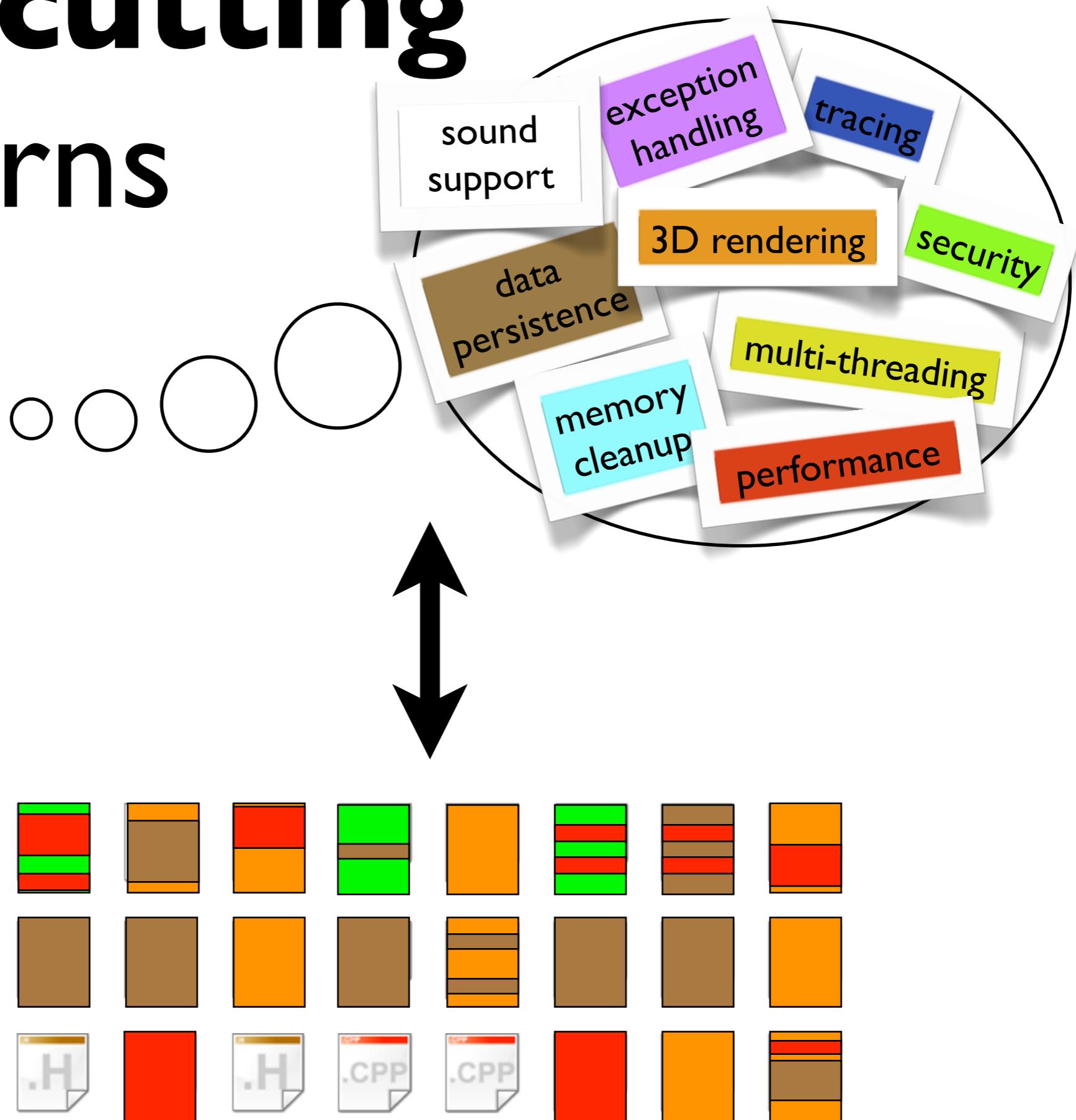
# Crosscutting Concerns



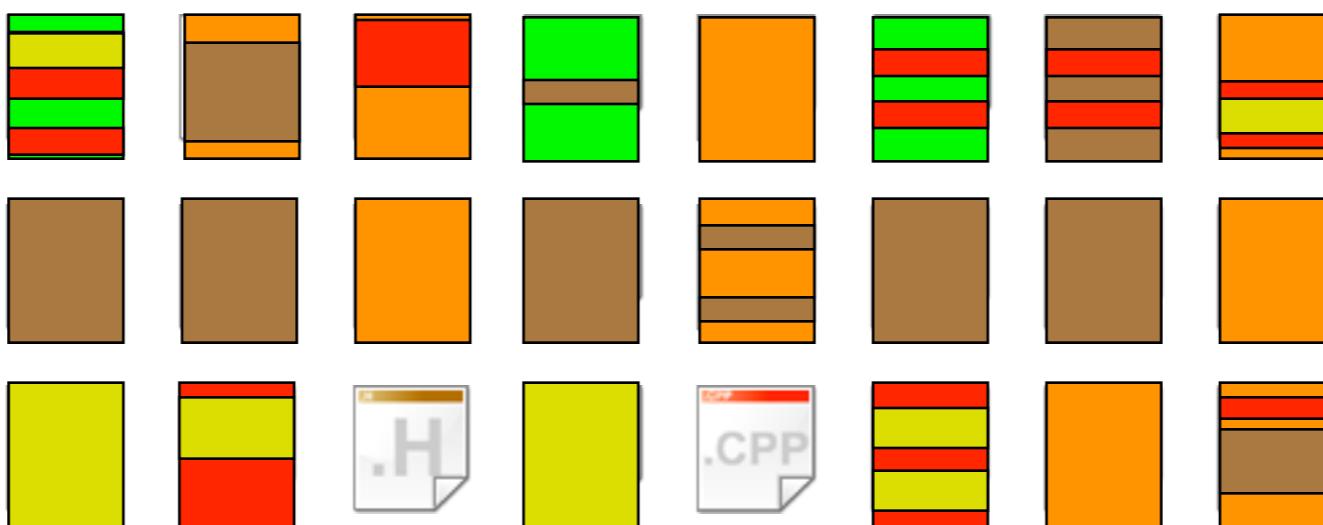
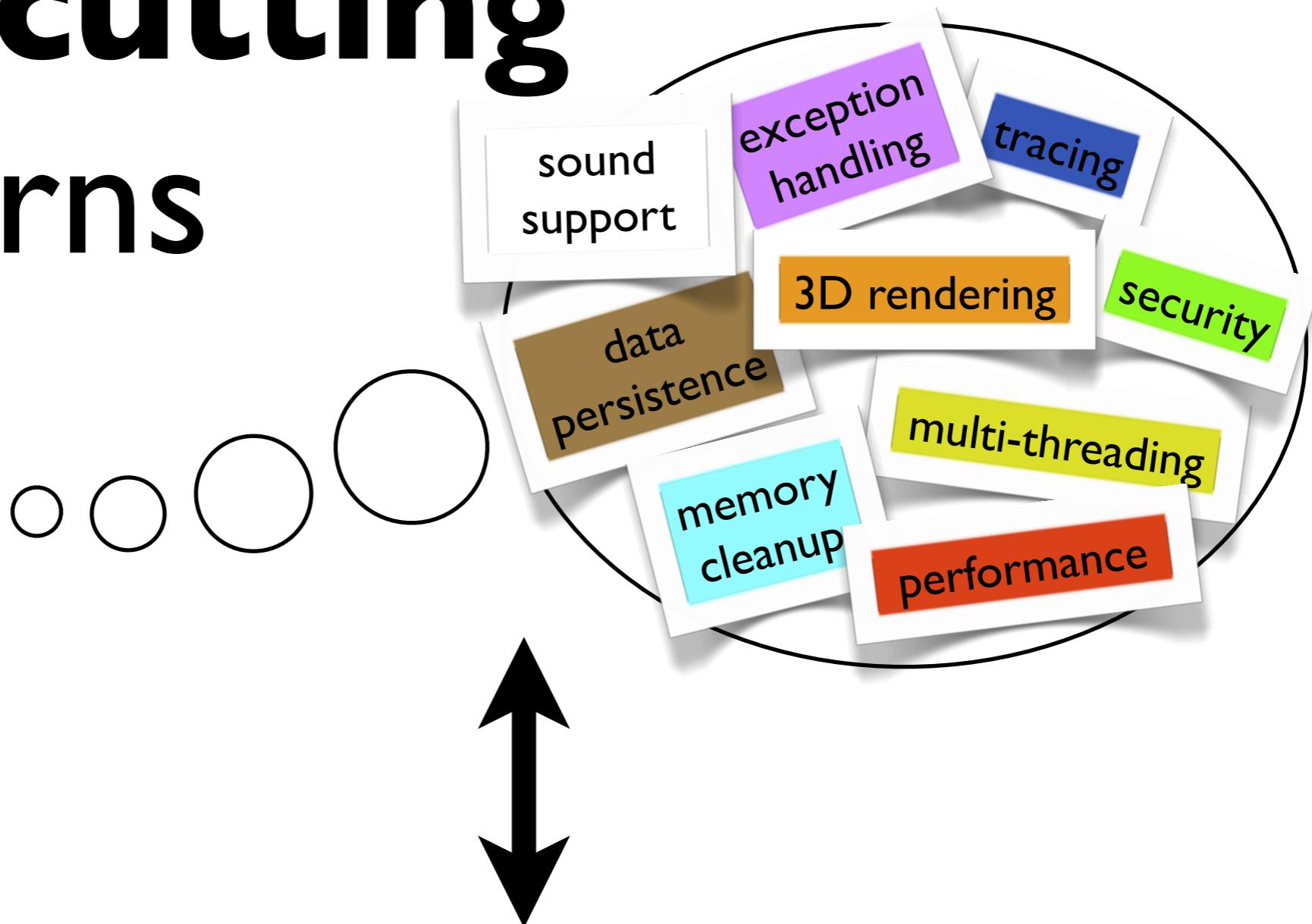
# Crosscutting Concerns



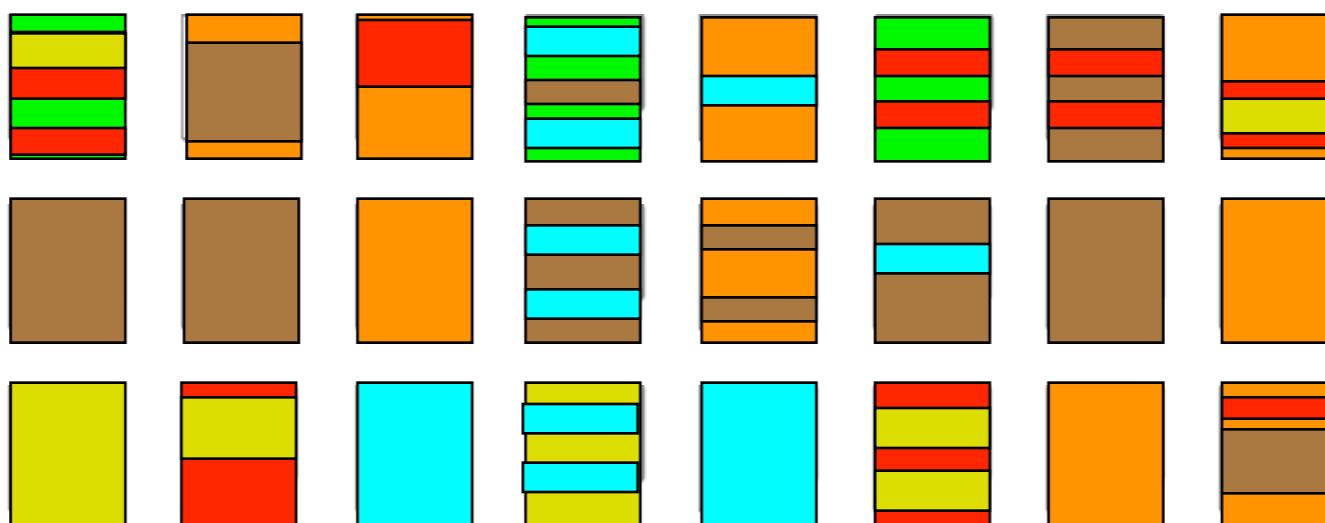
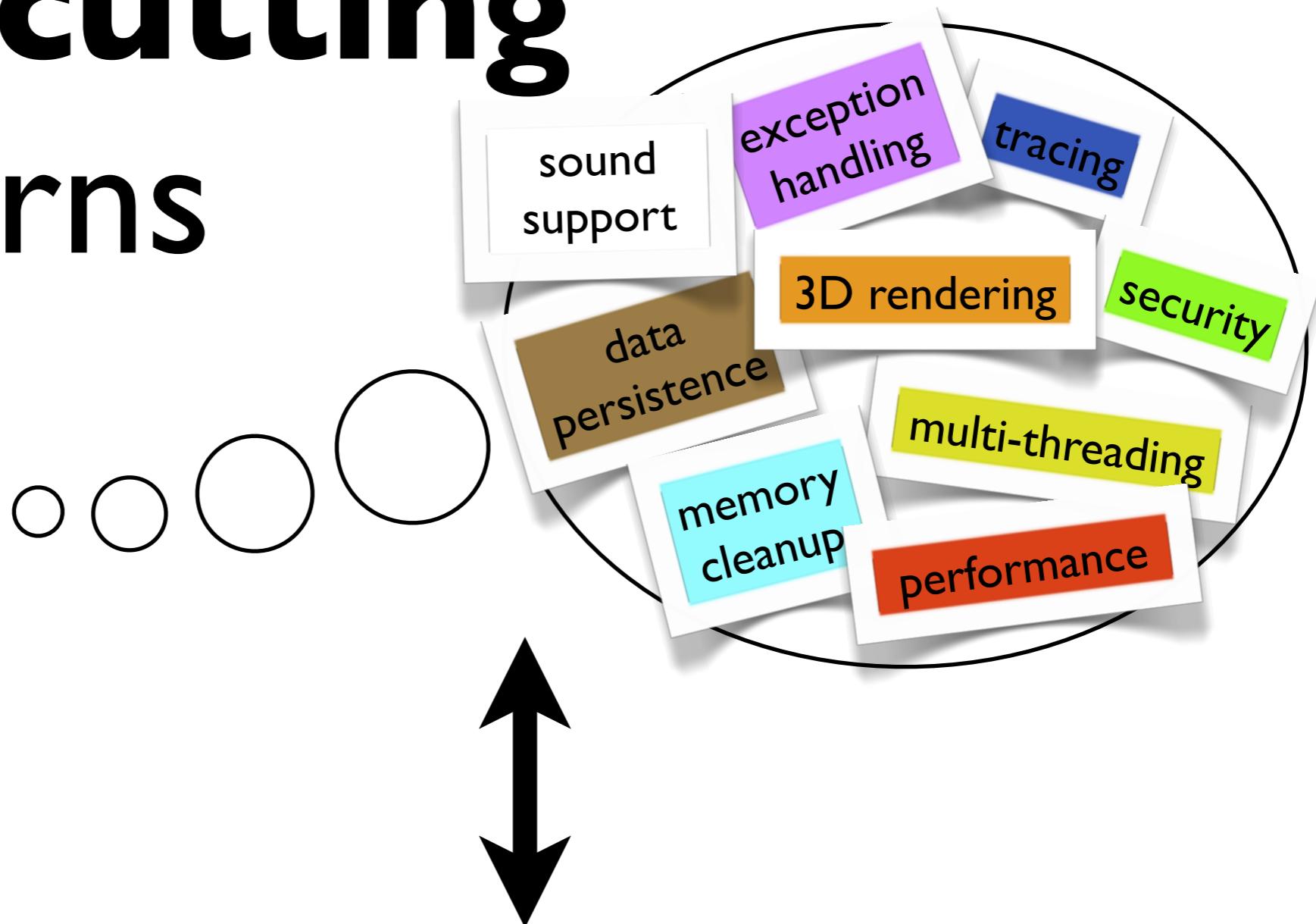
# Crosscutting Concerns



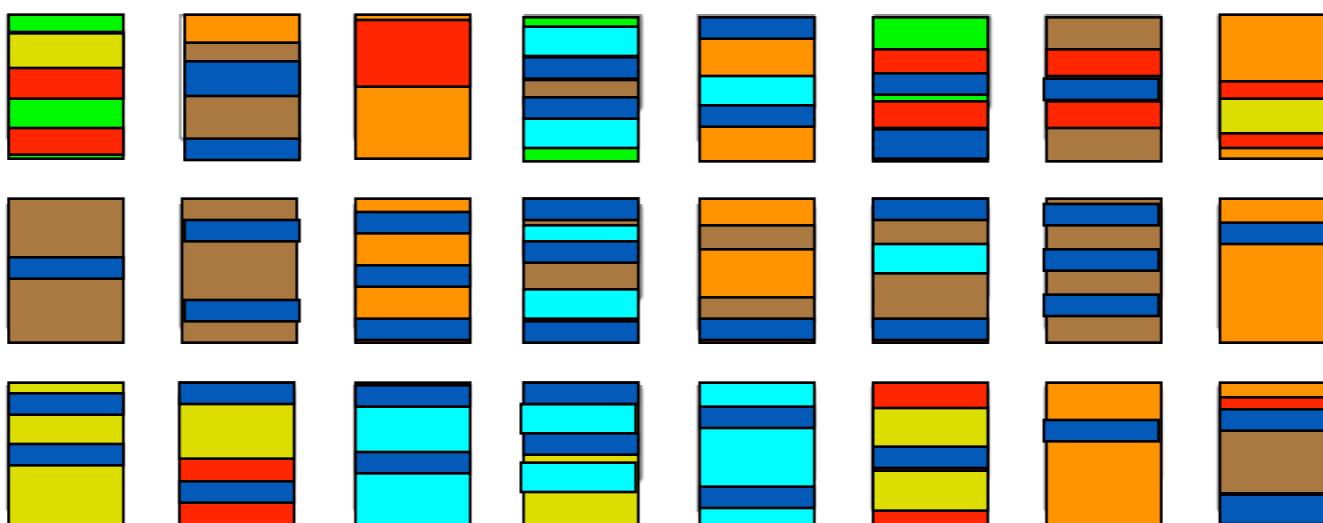
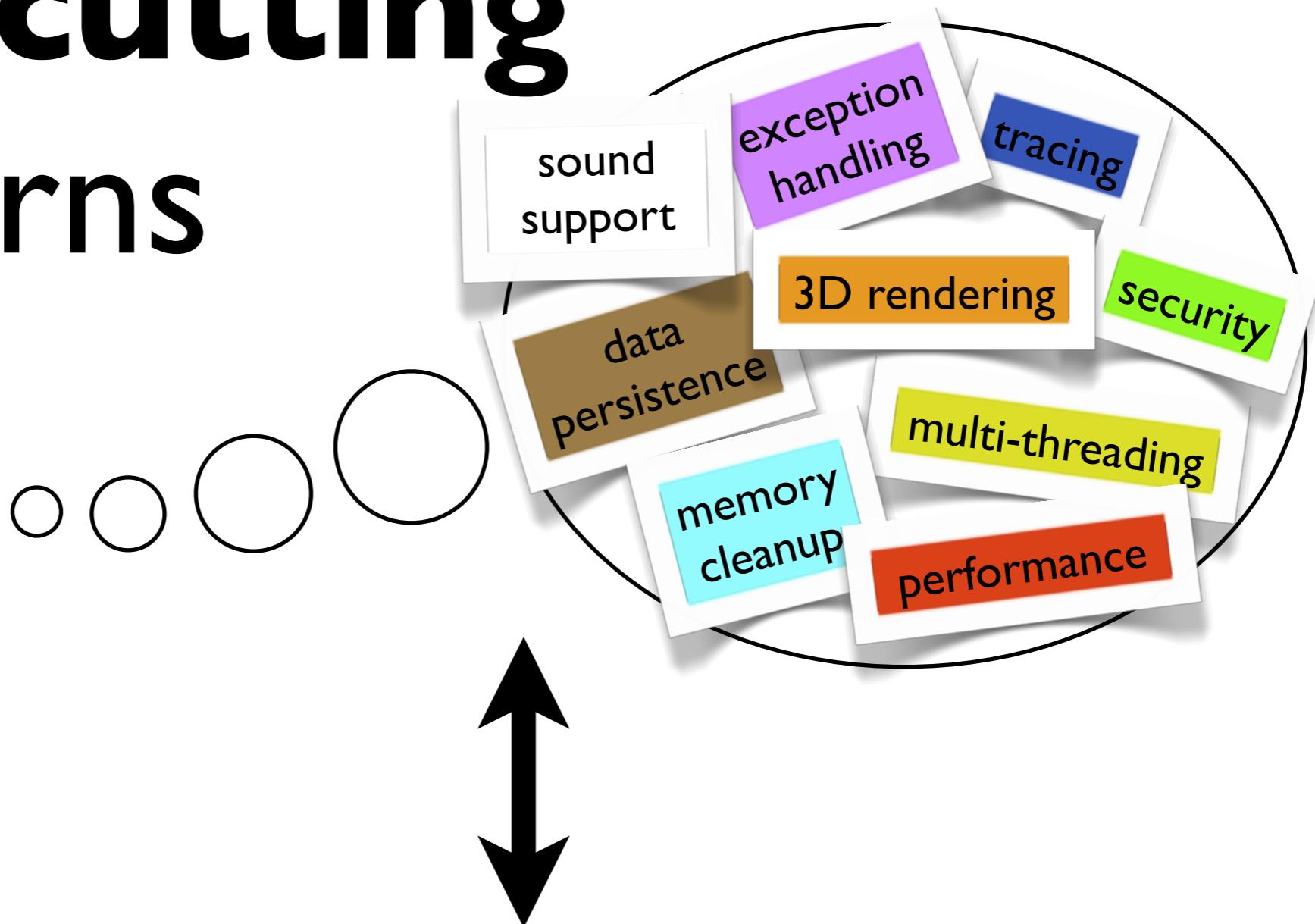
# Crosscutting Concerns



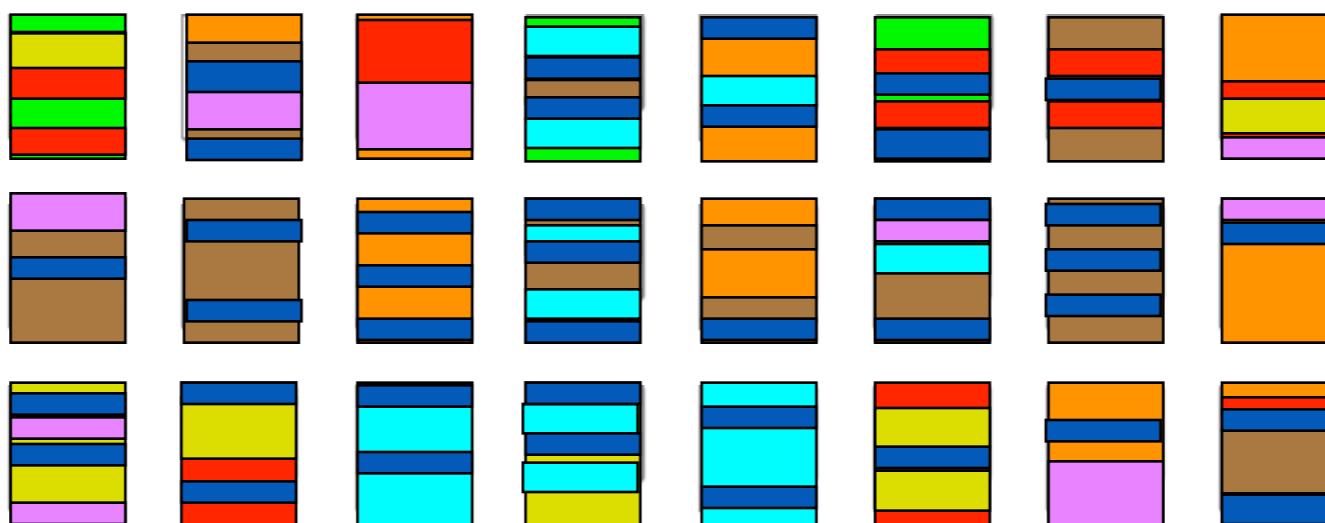
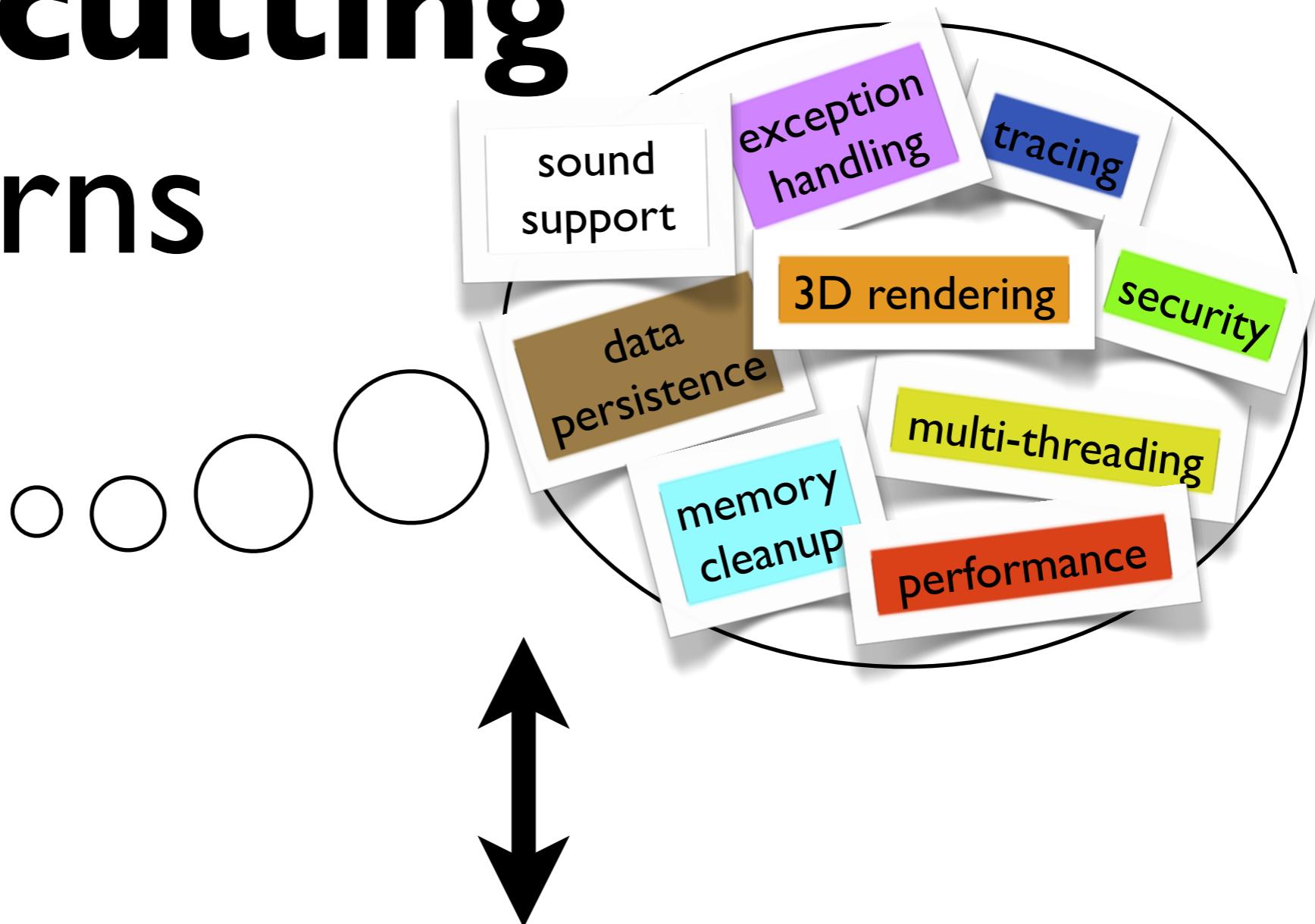
# Crosscutting Concerns



# Crosscutting Concerns



# Crosscutting Concerns



# (Crosscutting) Concern Mining

- I. Which** concerns are implemented?
- 2. Where?**
- 3. How** are concerns **composed** together?

# I. What is a Crosscutting Concern?

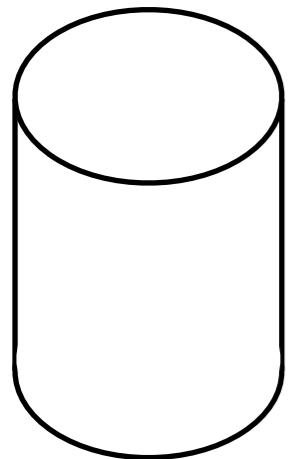
2. The Concern Mining Process and its Shortcomings

3. COMMIT

4. Case Study

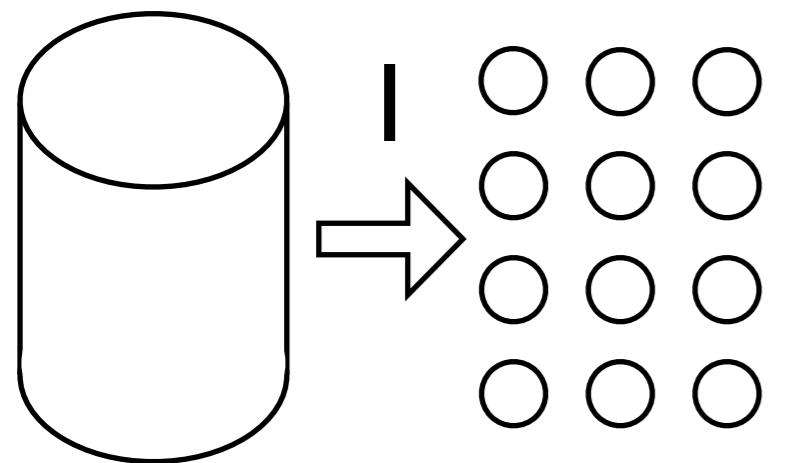
5. Conclusion

# Concern Mining Process



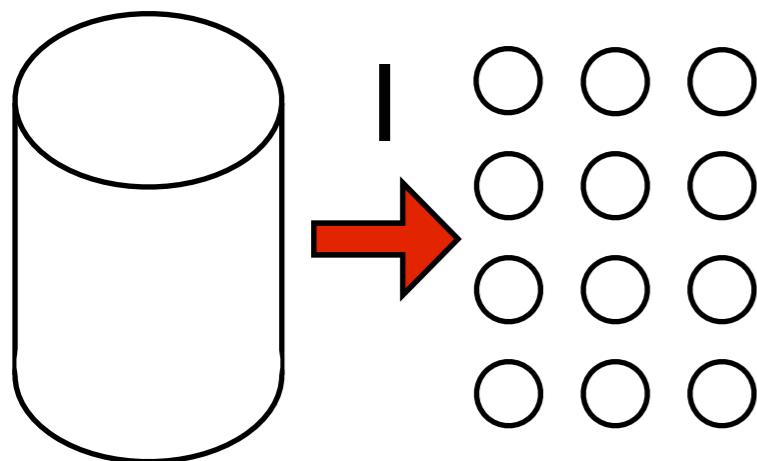
**data  
source**

# Concern Mining Process



**data source**      **concern seeds**

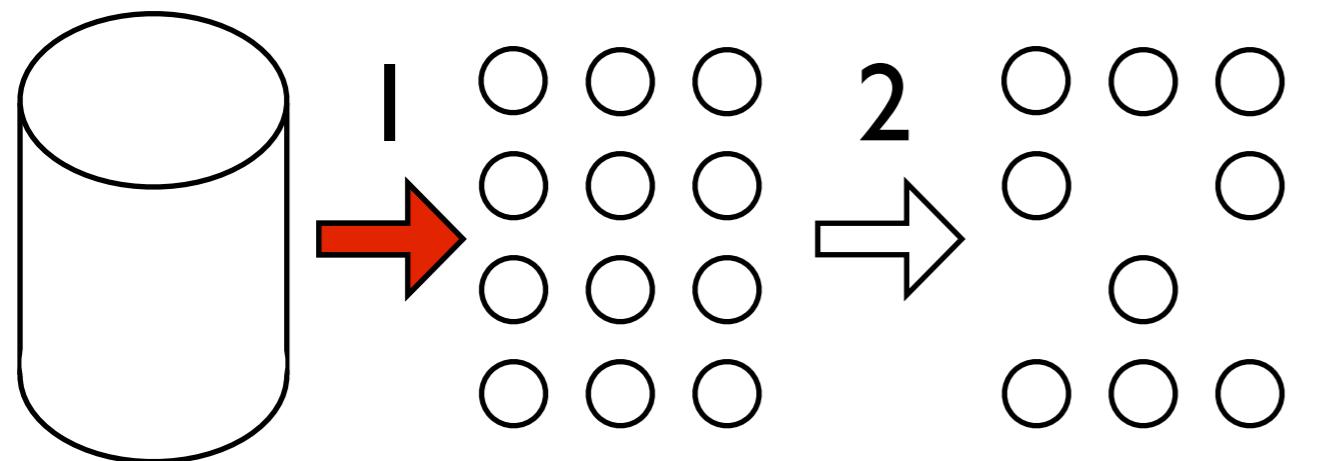
# Concern Mining Process



data concern  
source seeds

**Concern Mining  
Techniques :-)**

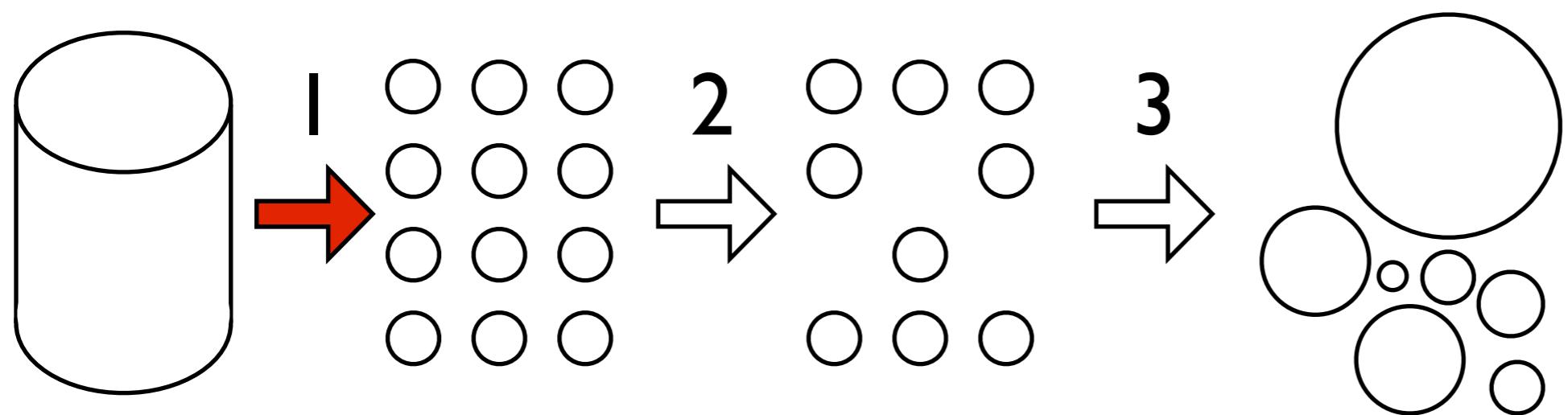
# Concern Mining Process



data concern concerns  
source seeds

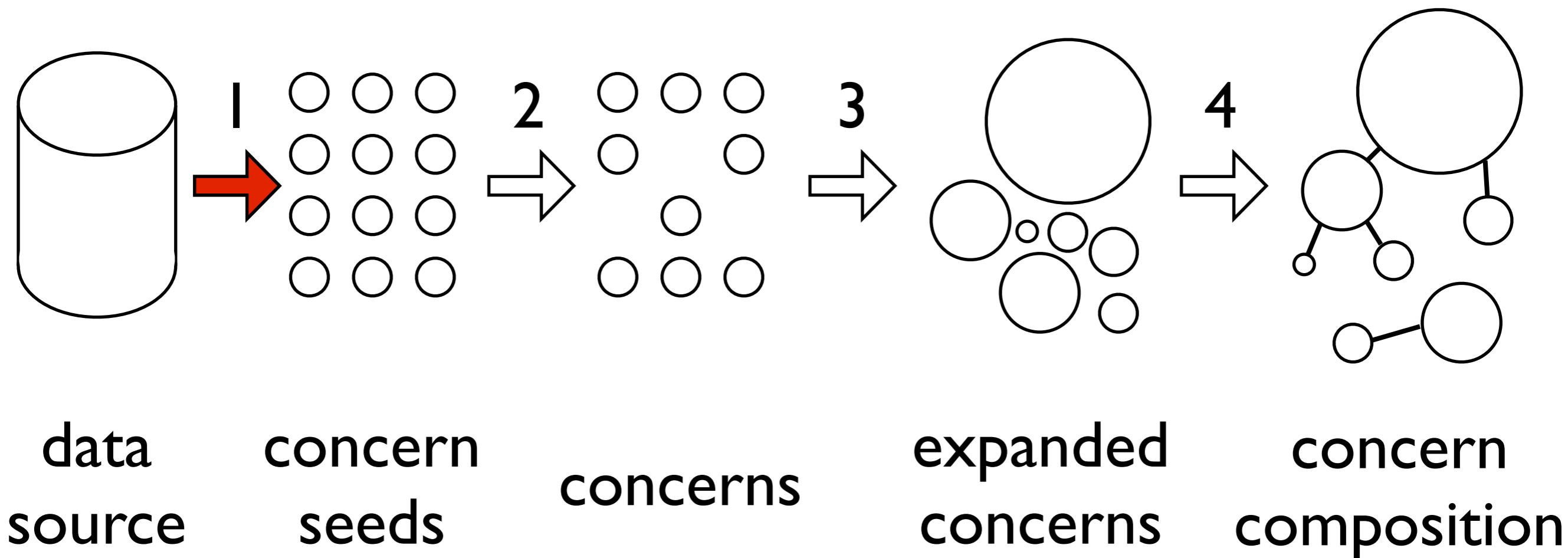
**Concern Mining  
Techniques :-)**

# Concern Mining Process



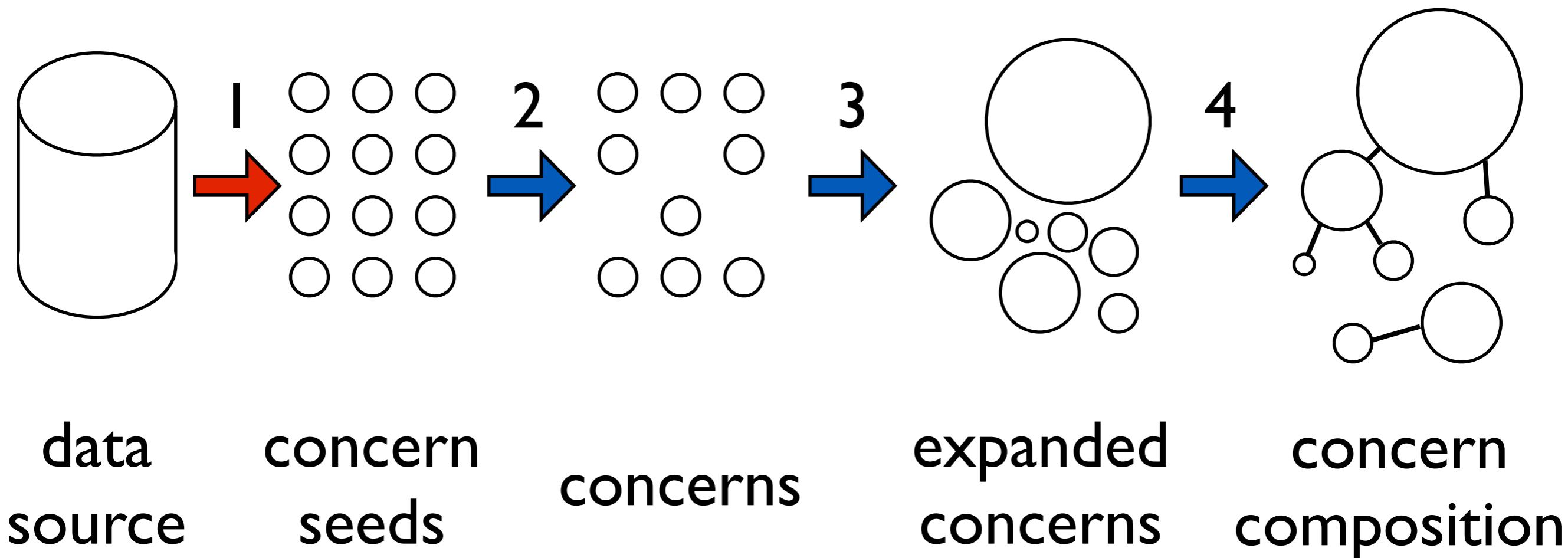
**Concern Mining  
Techniques :-)**

# Concern Mining Process



**Concern Mining  
Techniques :-)**

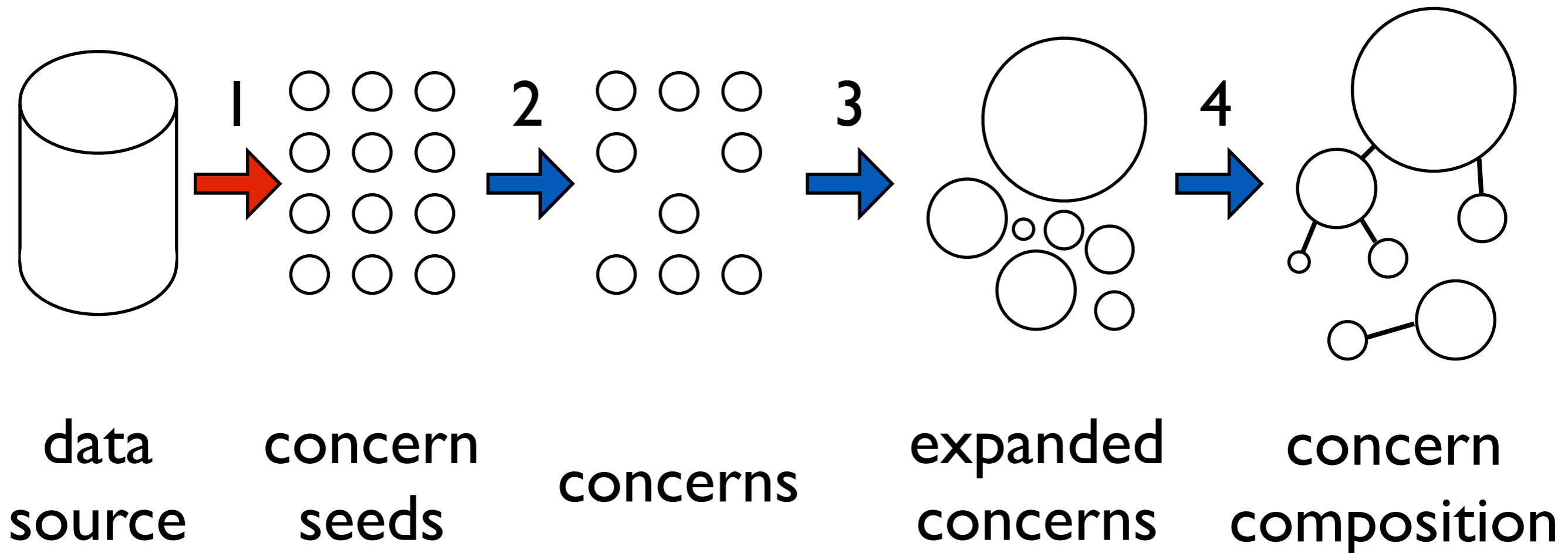
# Concern Mining Process

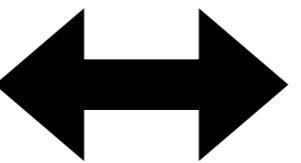


**Concern Mining  
Techniques :-)**

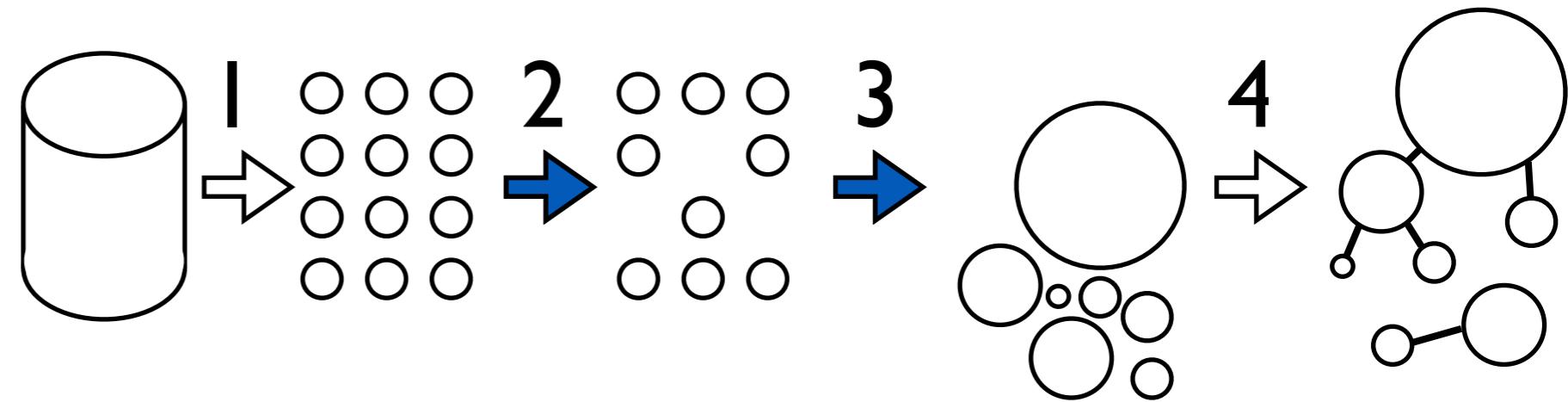
**MANUAL :-**

# Concern Mining Process

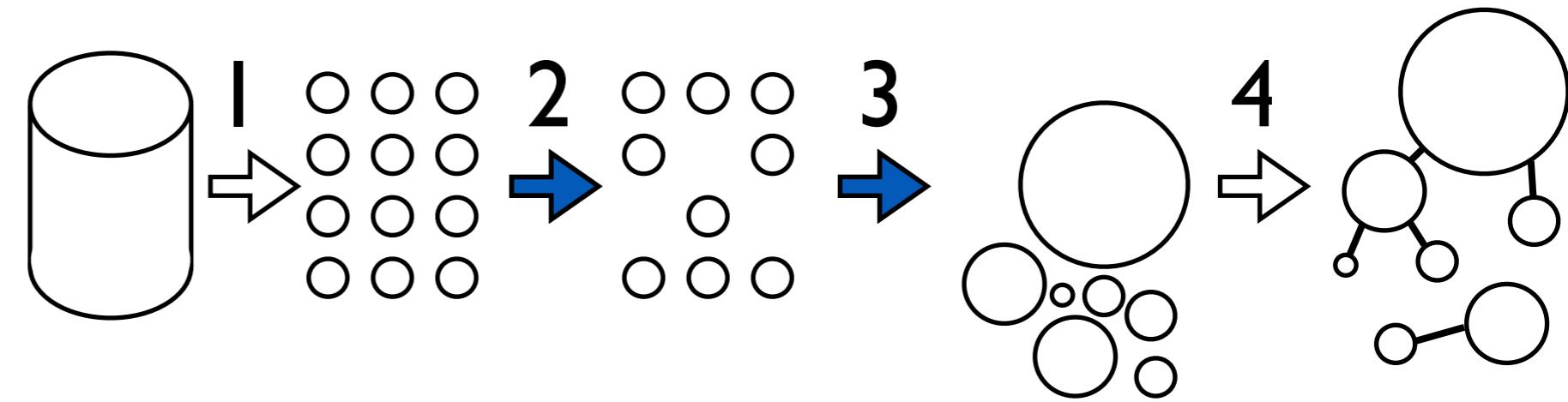


**Concern Mining  
Techniques :-)**  **MANUAL :-**

# SI: Limited Context



# SI: Limited Context

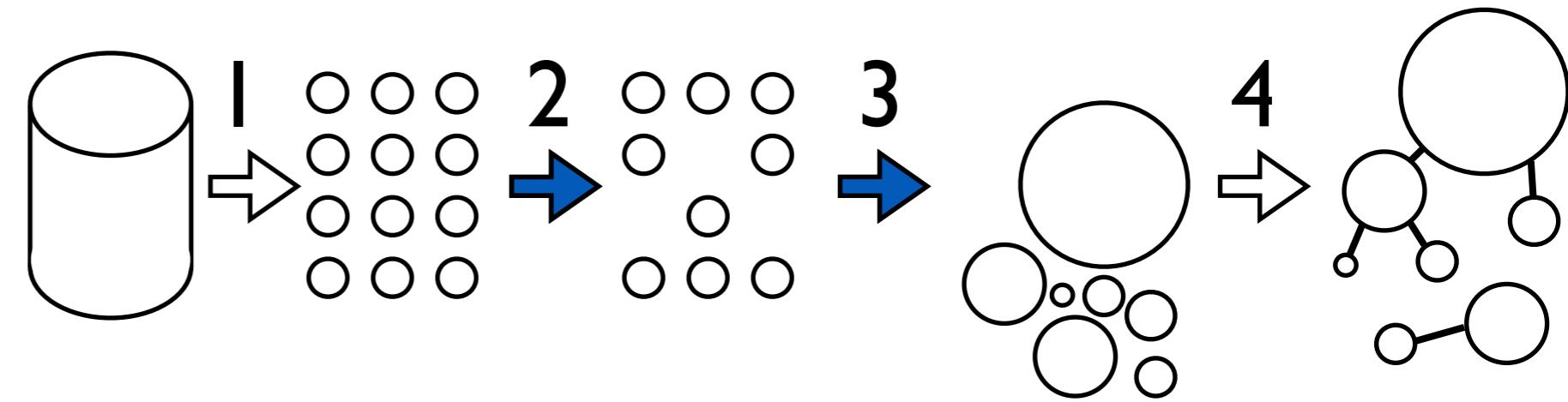


`block()`

`thread()`

`process()`  
`clean()`

# SI: Limited Context



**DEFINED\_LINUX**

`semaphore_t`

`block()`

`thread()`

`process()`

`clean()`

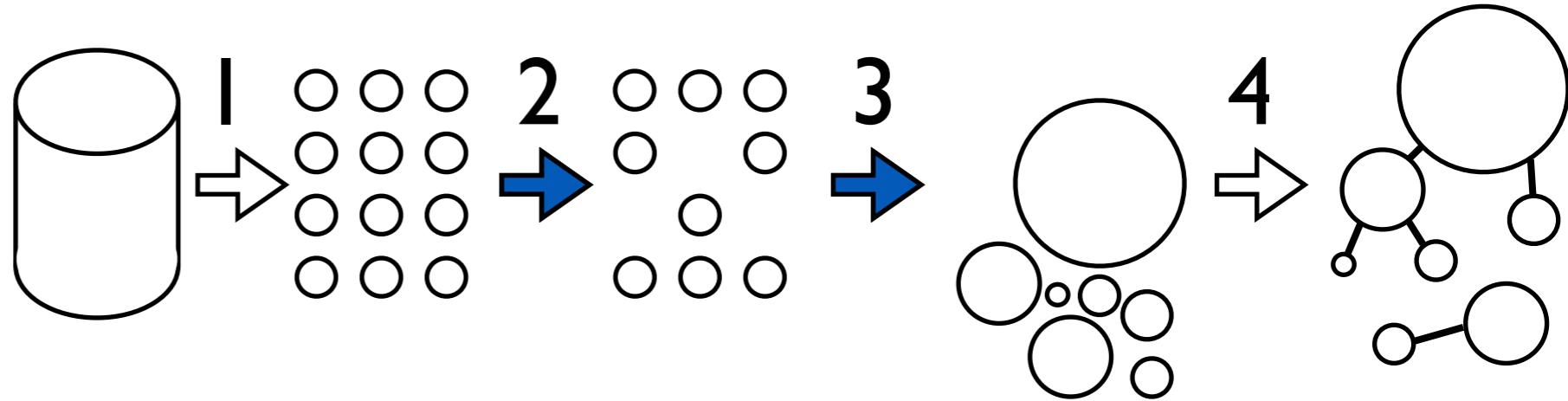
`address`

`sender`

`subject`



# SI: Limited Context



`semaphore_t`

`DEFINED_LINUX`

`block()`

`mutex`

`thread()`

`clean()`

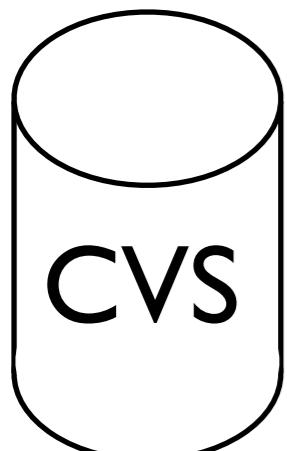
`process()`

`address`

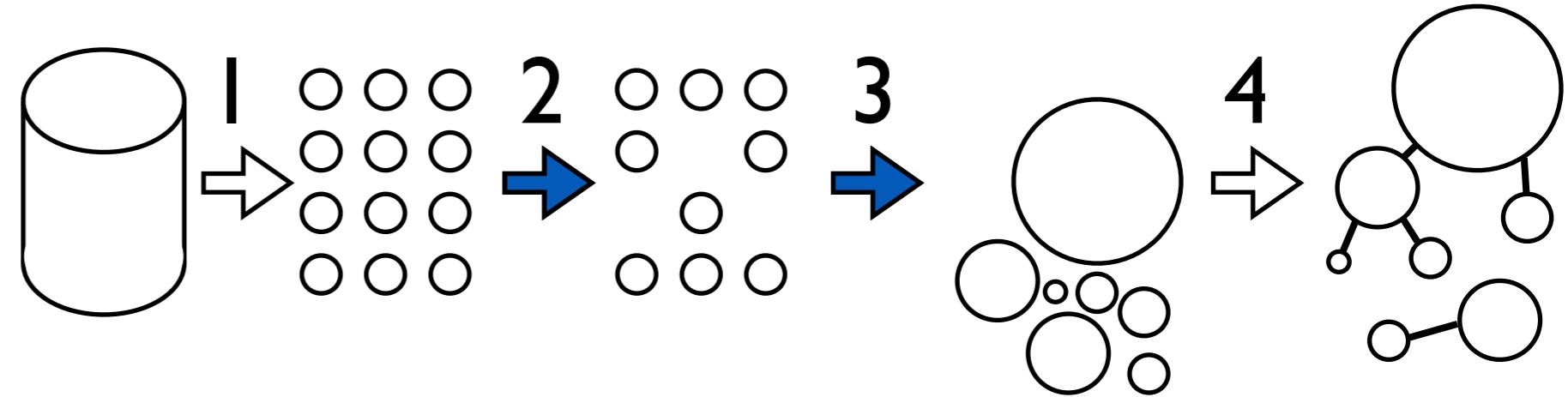
`sender`

`subject`

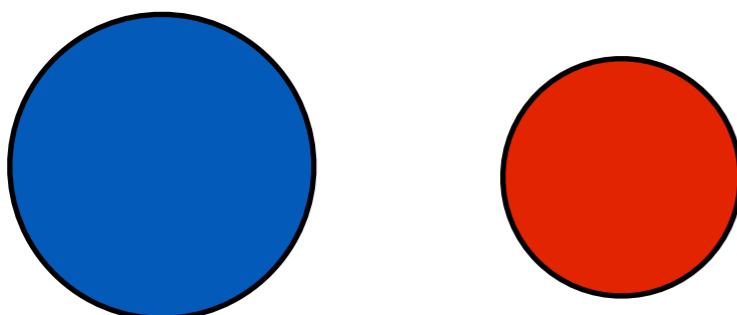
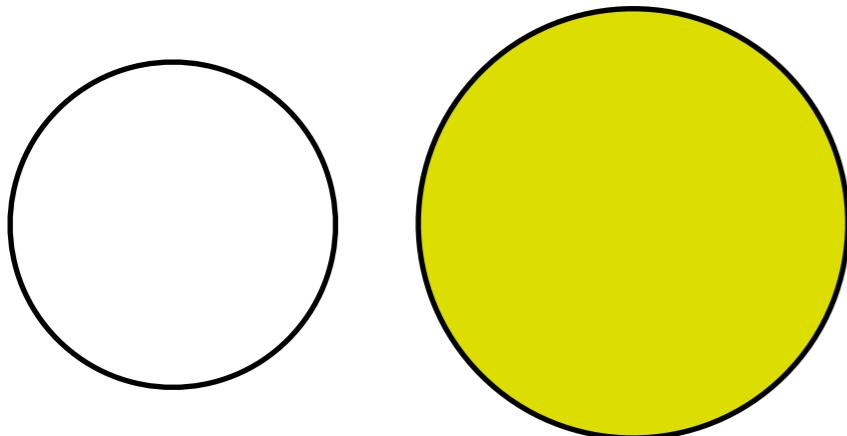
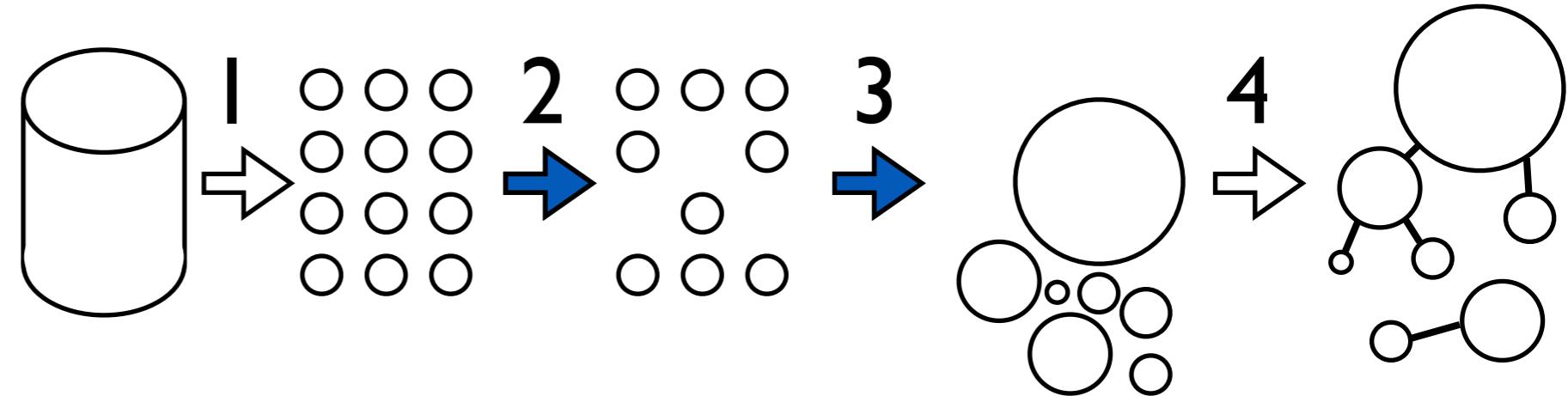
`thread()`



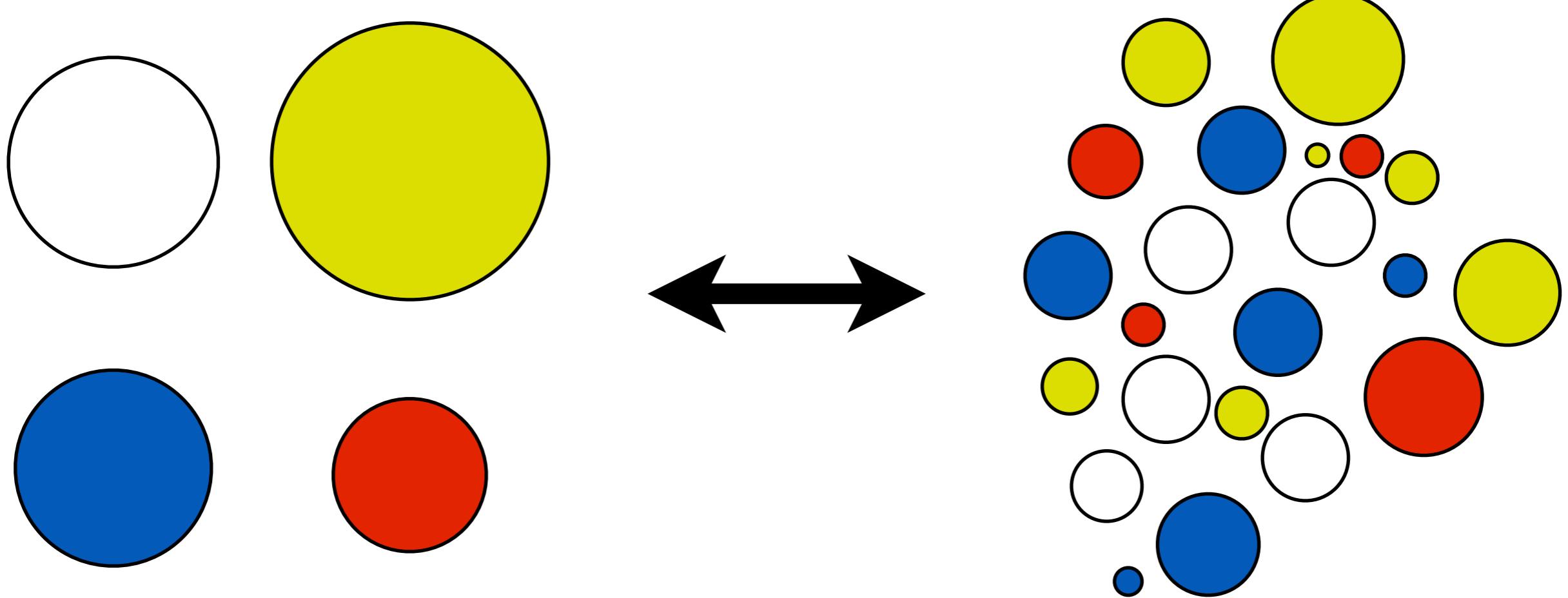
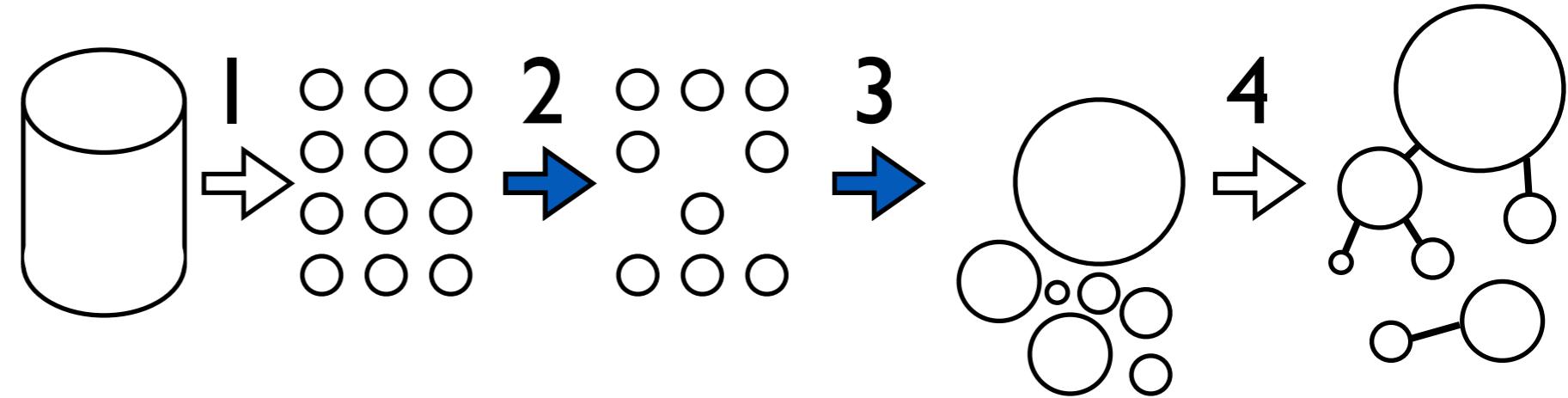
## S2: Noise



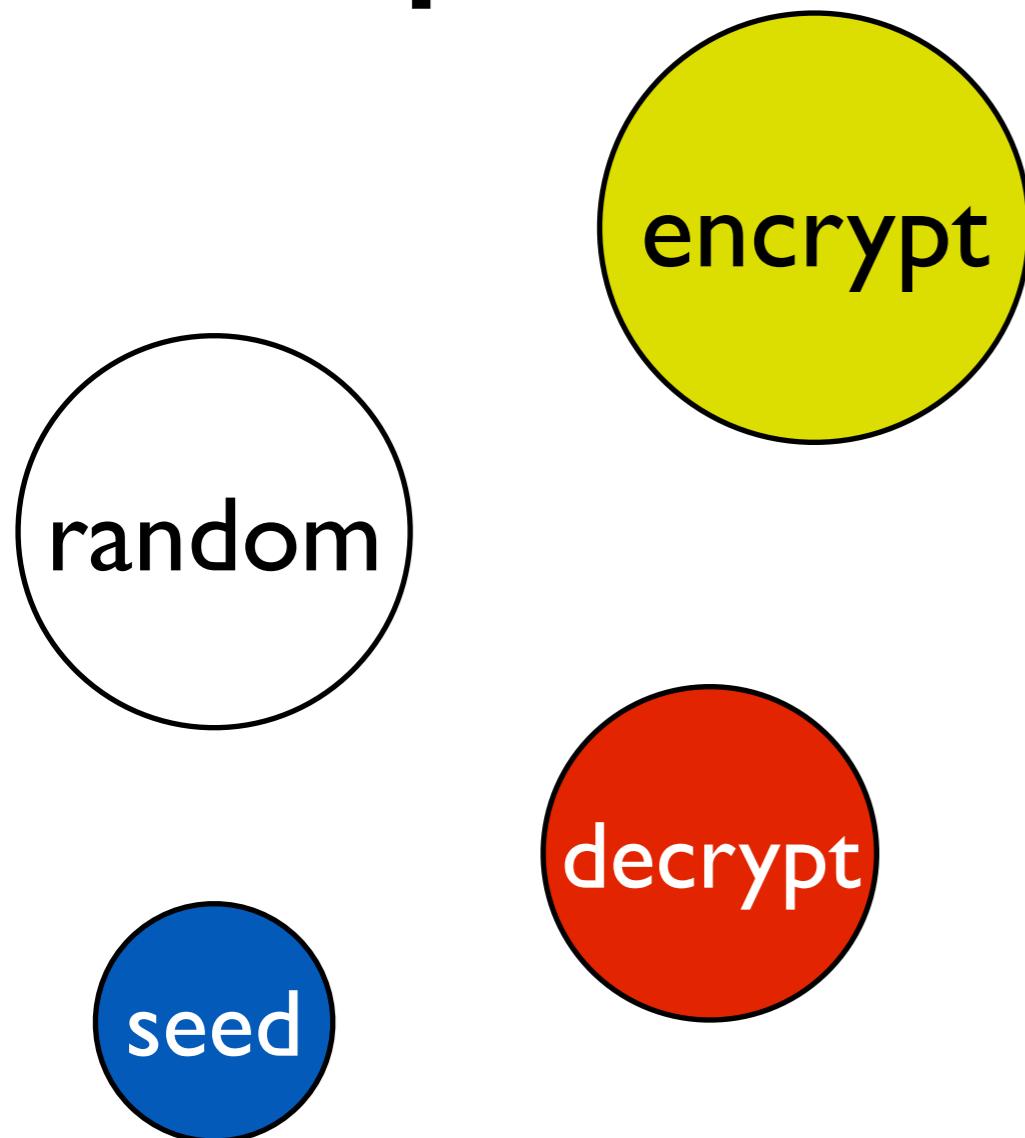
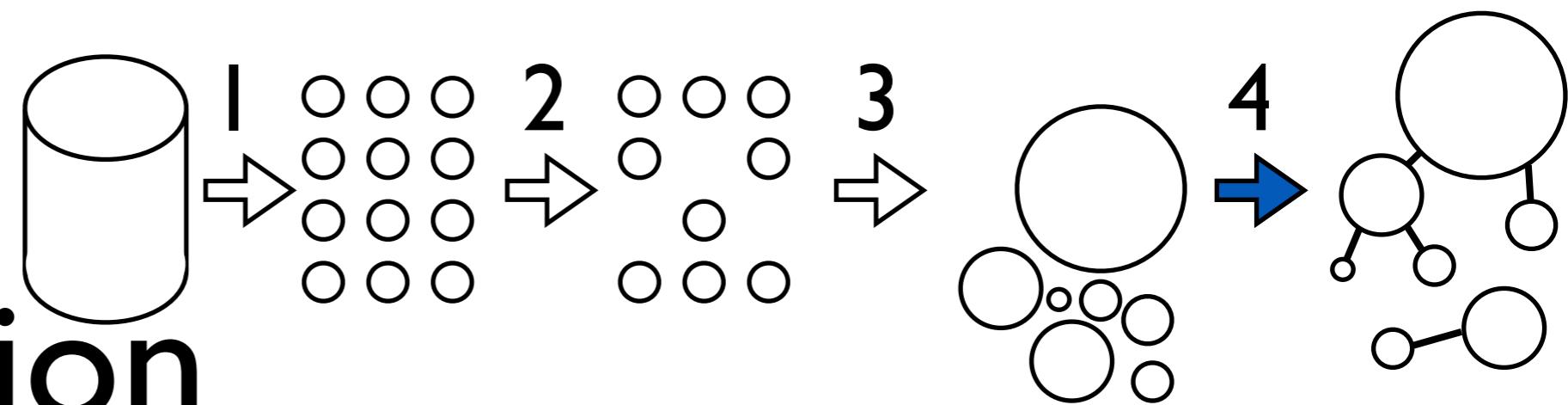
## S2: Noise



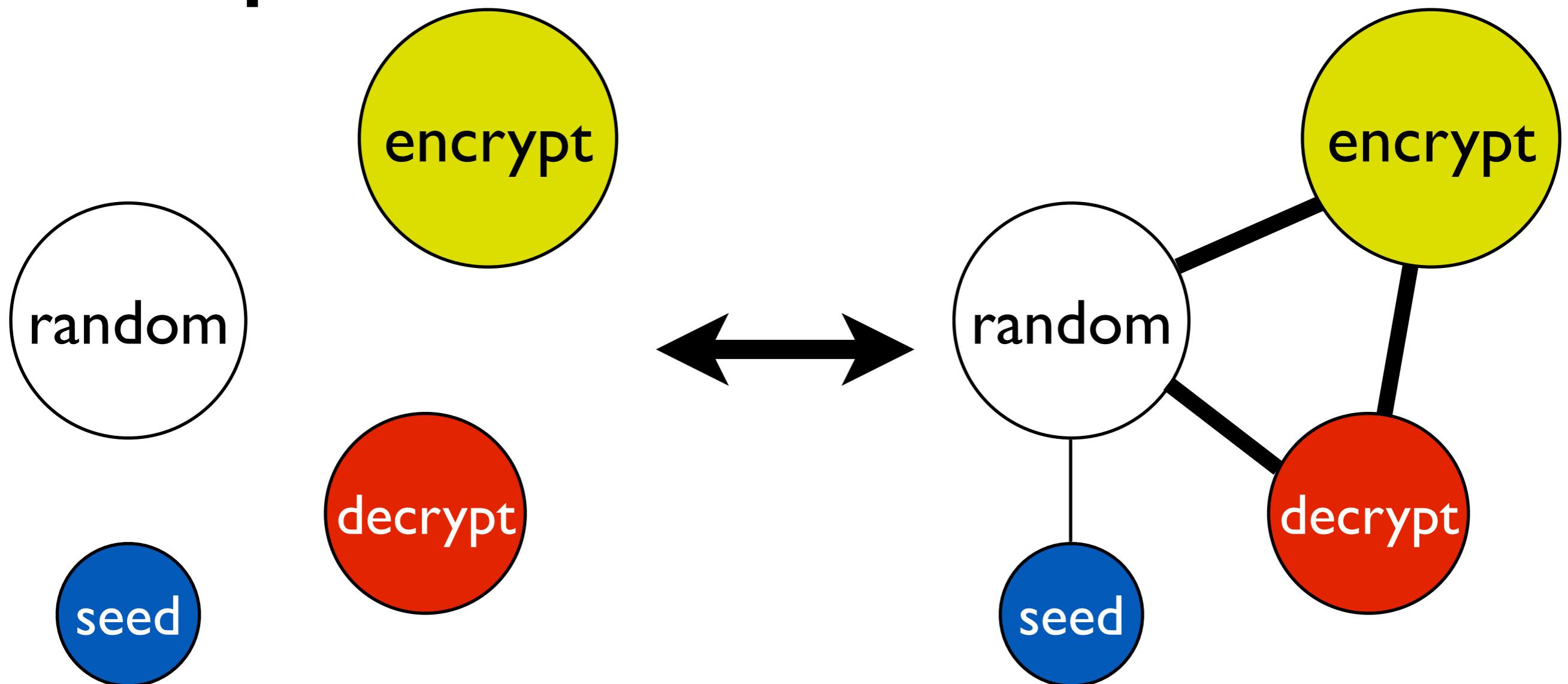
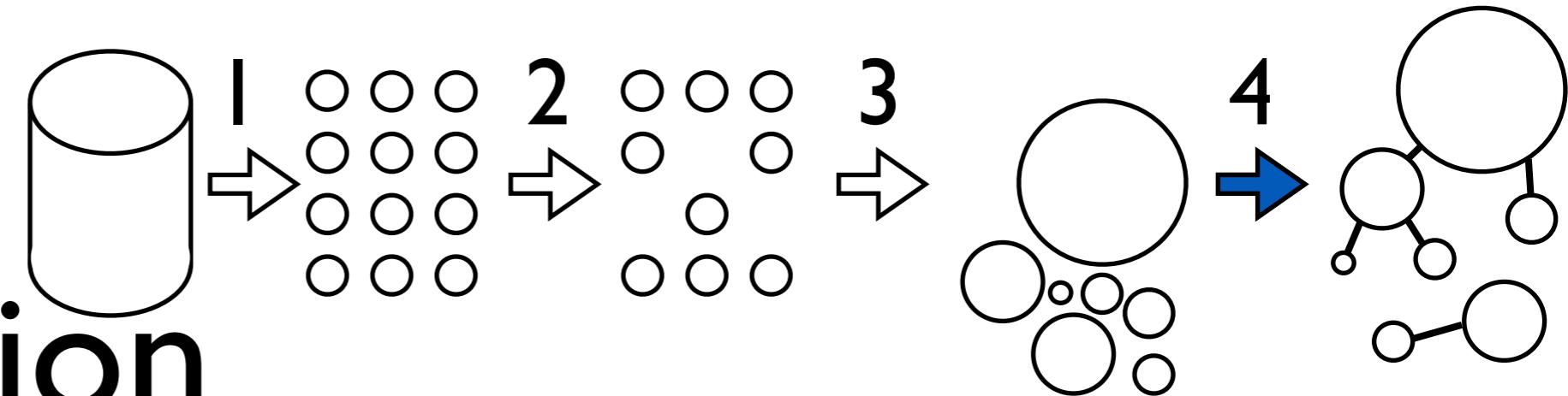
## S2: Noise



# S3: No Composition

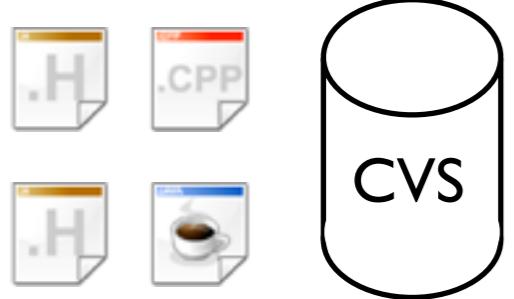


# S3: No Composition

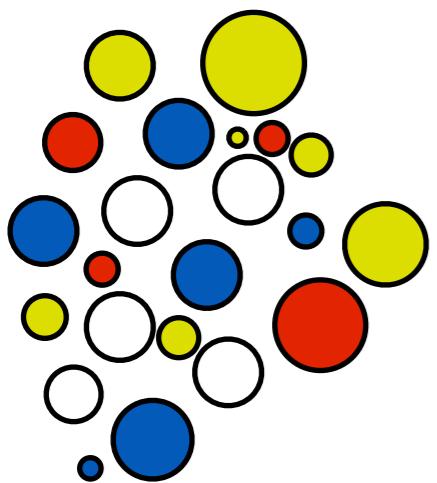


- 1. What is a Crosscutting Concern?**
- 2. The Concern Mining Process and its Shortcomings**
- 3. COMMIT**
- 4. Case Study**
- 5. Conclusion**

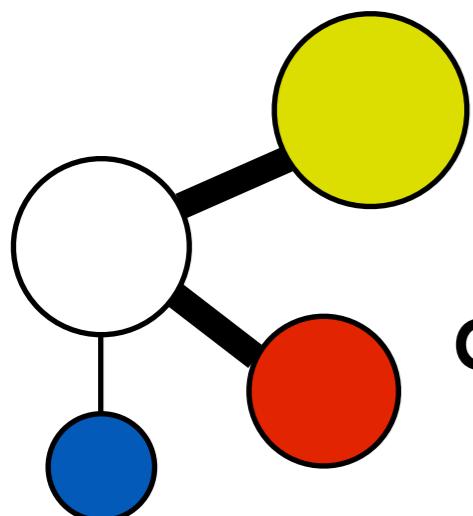
# Concern Mining using Mutual Information over Time



limited  
context



noise

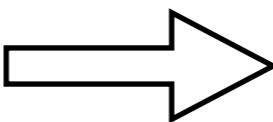


no  
composition

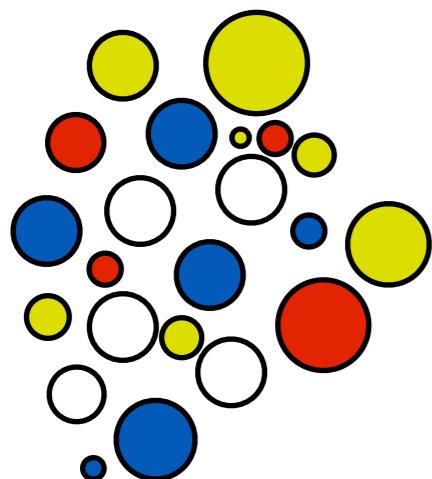
# Concern Mining using Mutual Information over Time



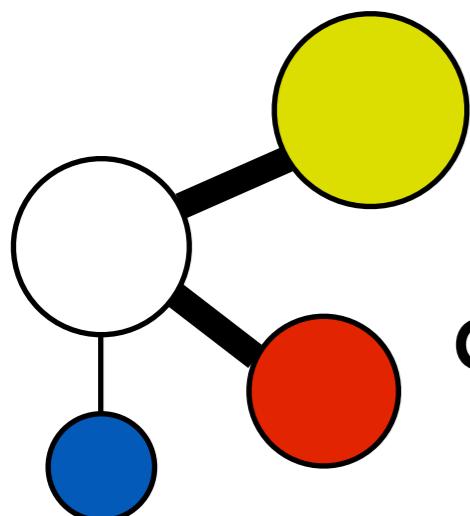
limited context



analyze **historical** changes to **all** code entities



noise



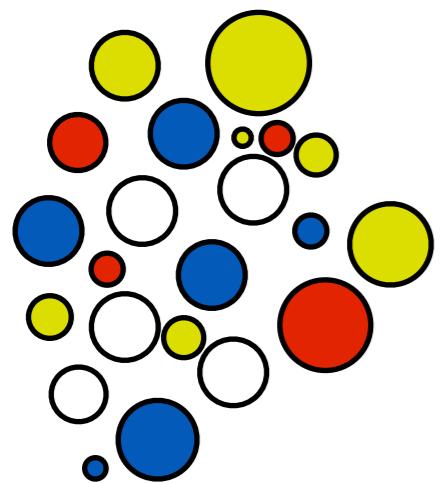
no  
composition

# Concern Mining using Mutual Information over Time

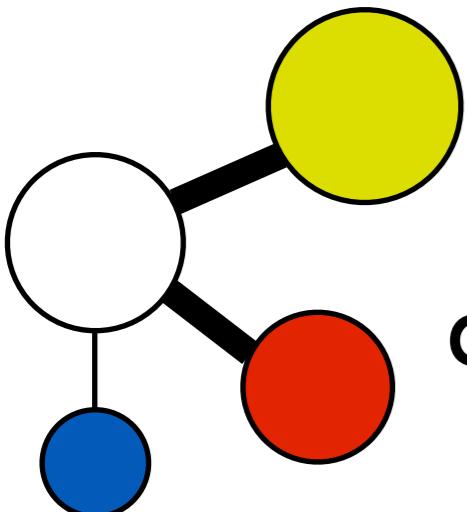
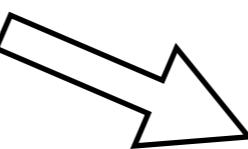


limited  
context

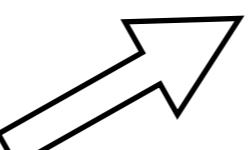
analyze **historical**  
changes to **all** code  
entities



noise

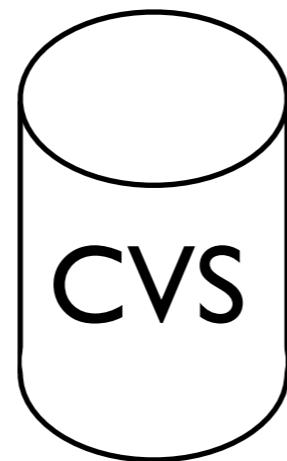


no  
composition

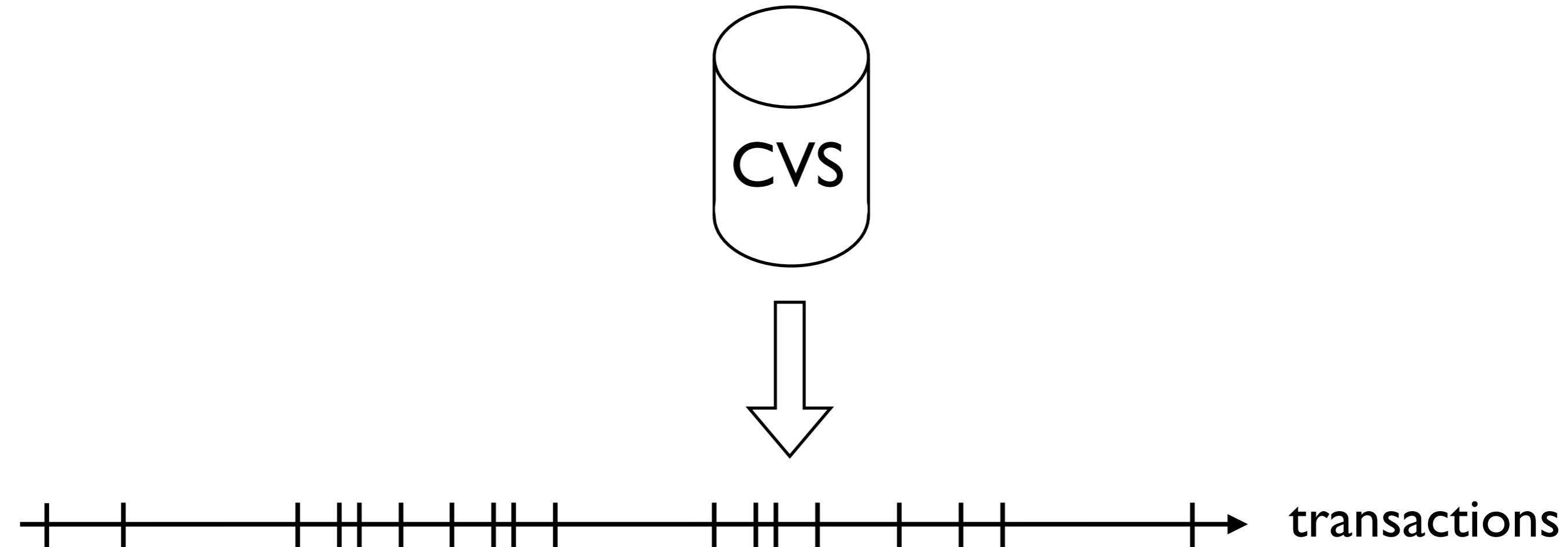
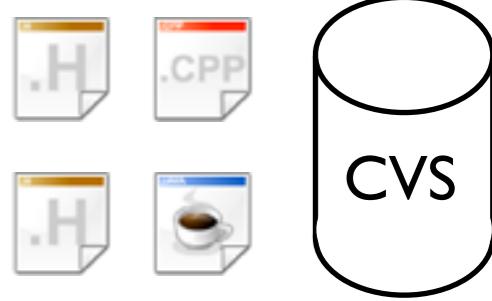


statistical clustering  
based on **mutual**  
**information**

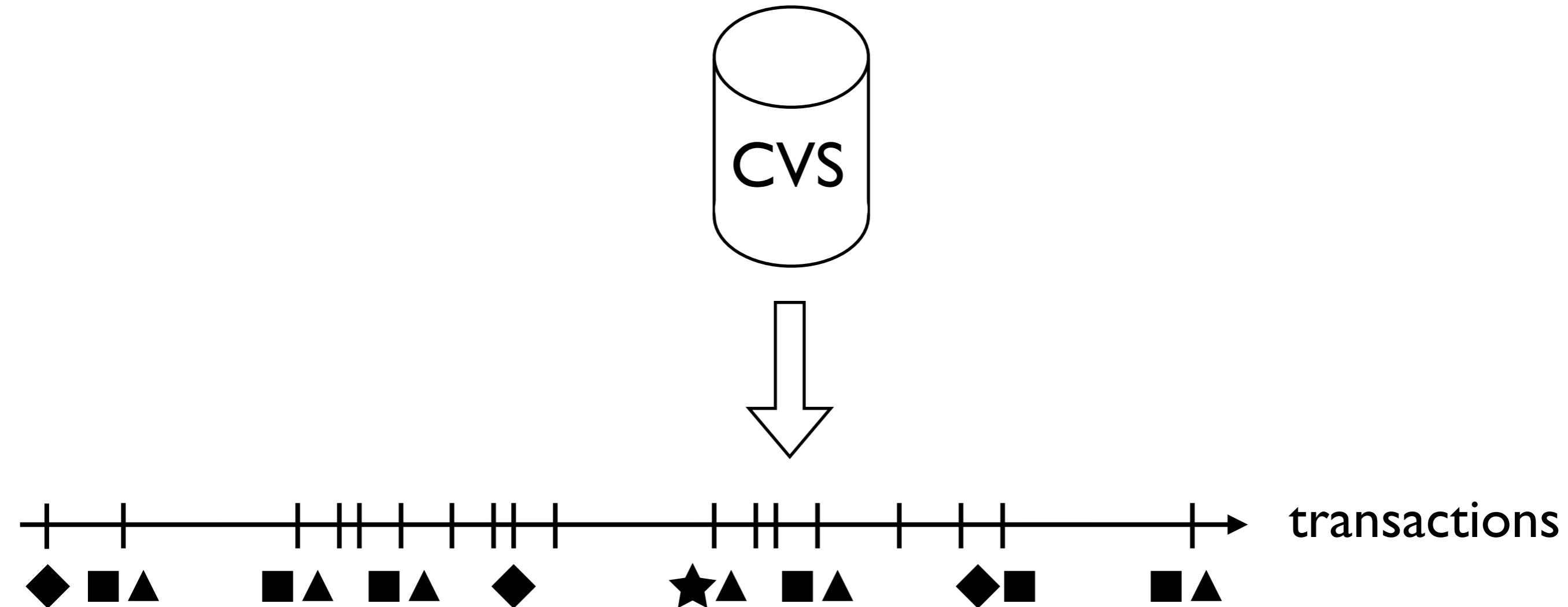
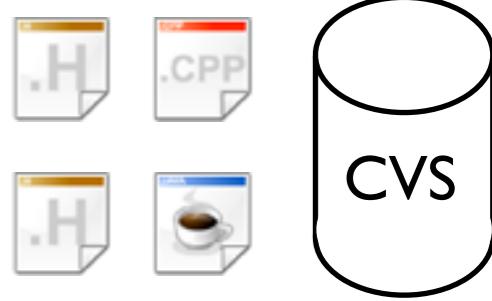
# SI. Historical Data Sources



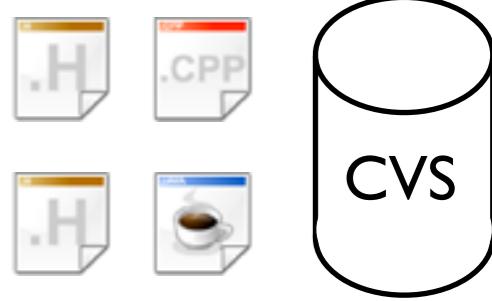
# SI. Historical Data Sources



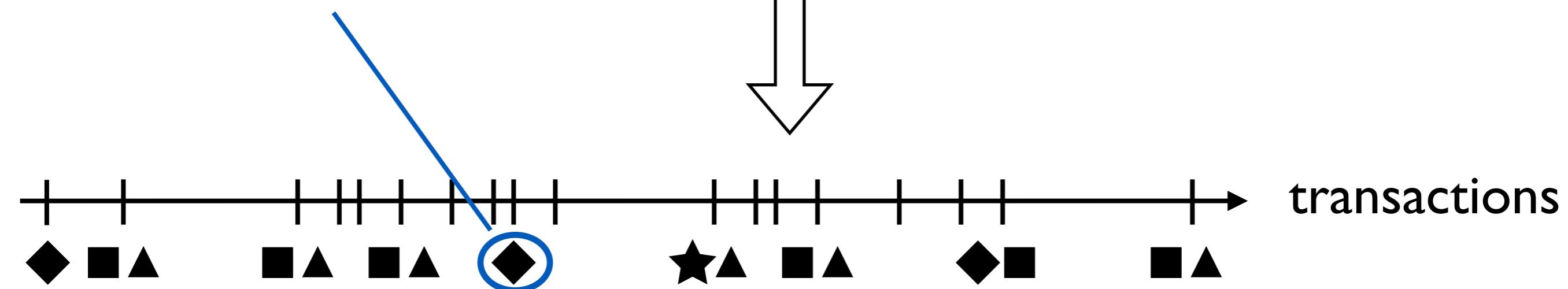
# SI. Historical Data Sources



# SI. Historical Data Sources



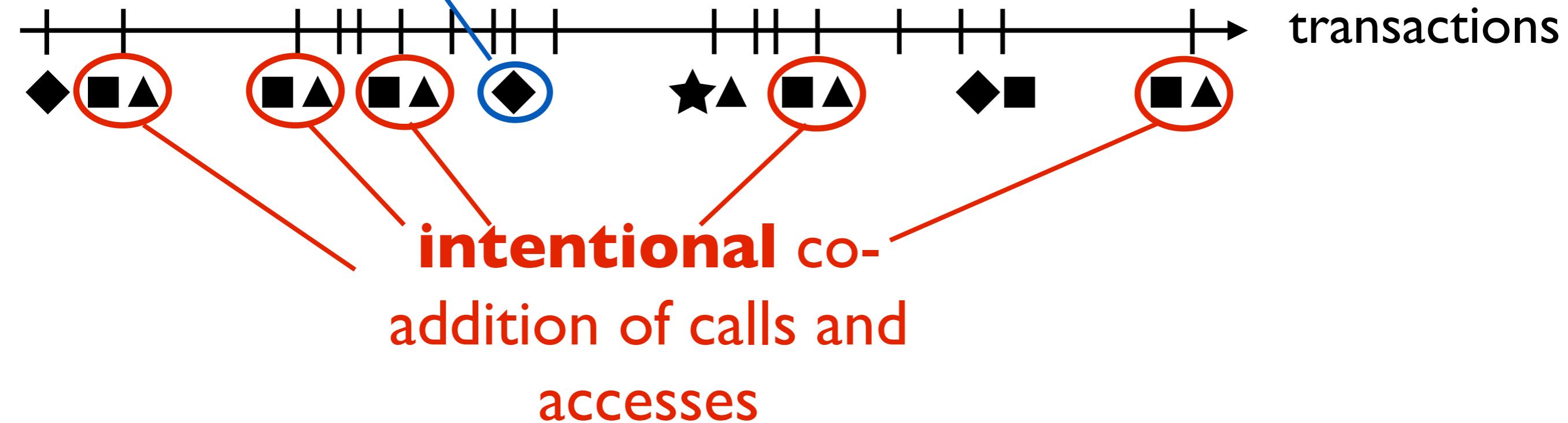
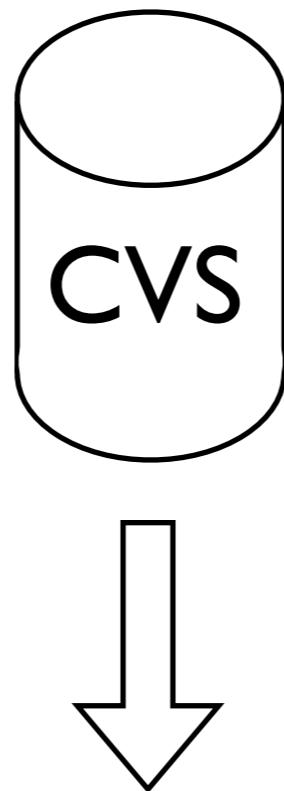
function call or  
variable access added



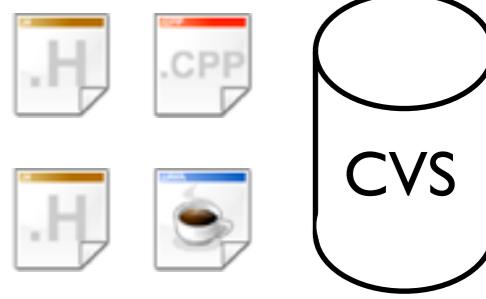
# SI. Historical Data Sources



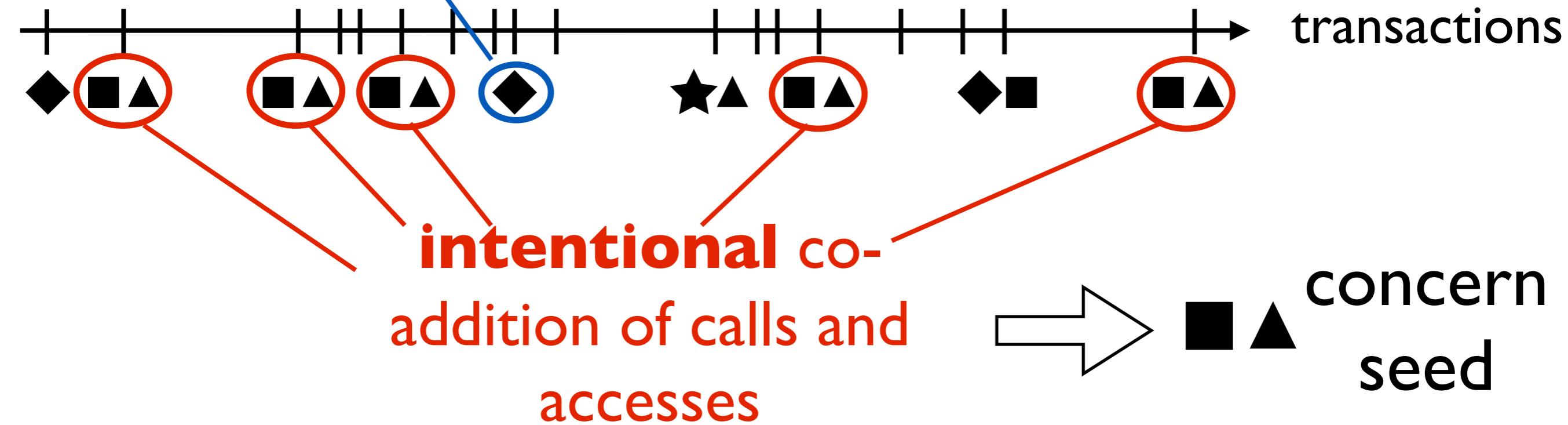
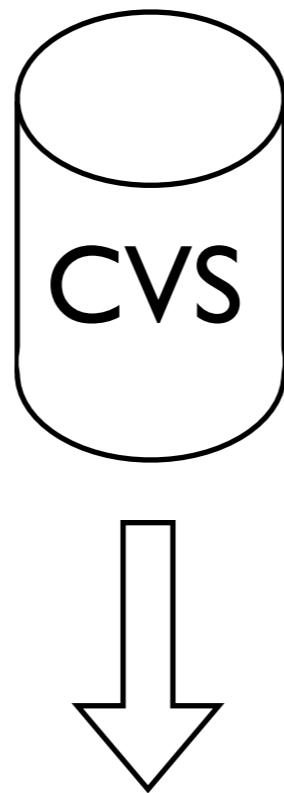
function call or  
variable access added

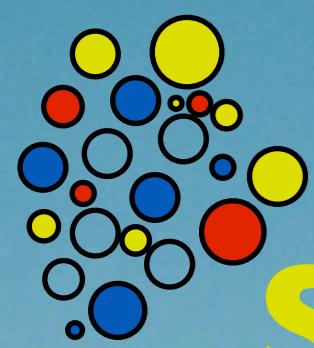


# SI. Historical Data Sources



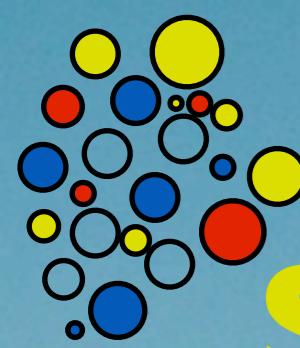
function call or  
variable access added





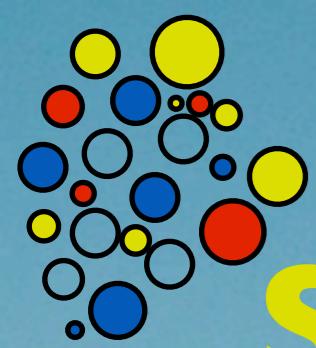
## S2. Mutual Information





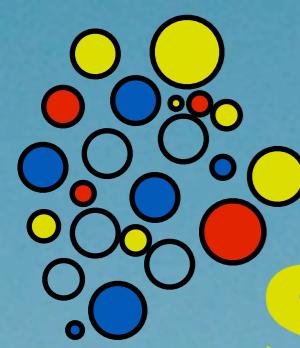
## S2. Mutual Information





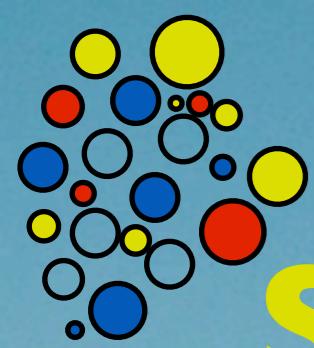
## S2. Mutual Information





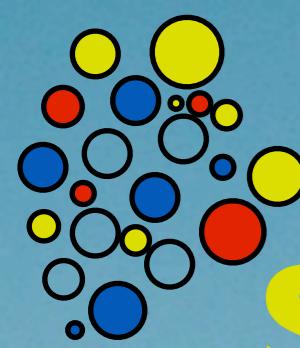
## S2. Mutual Information



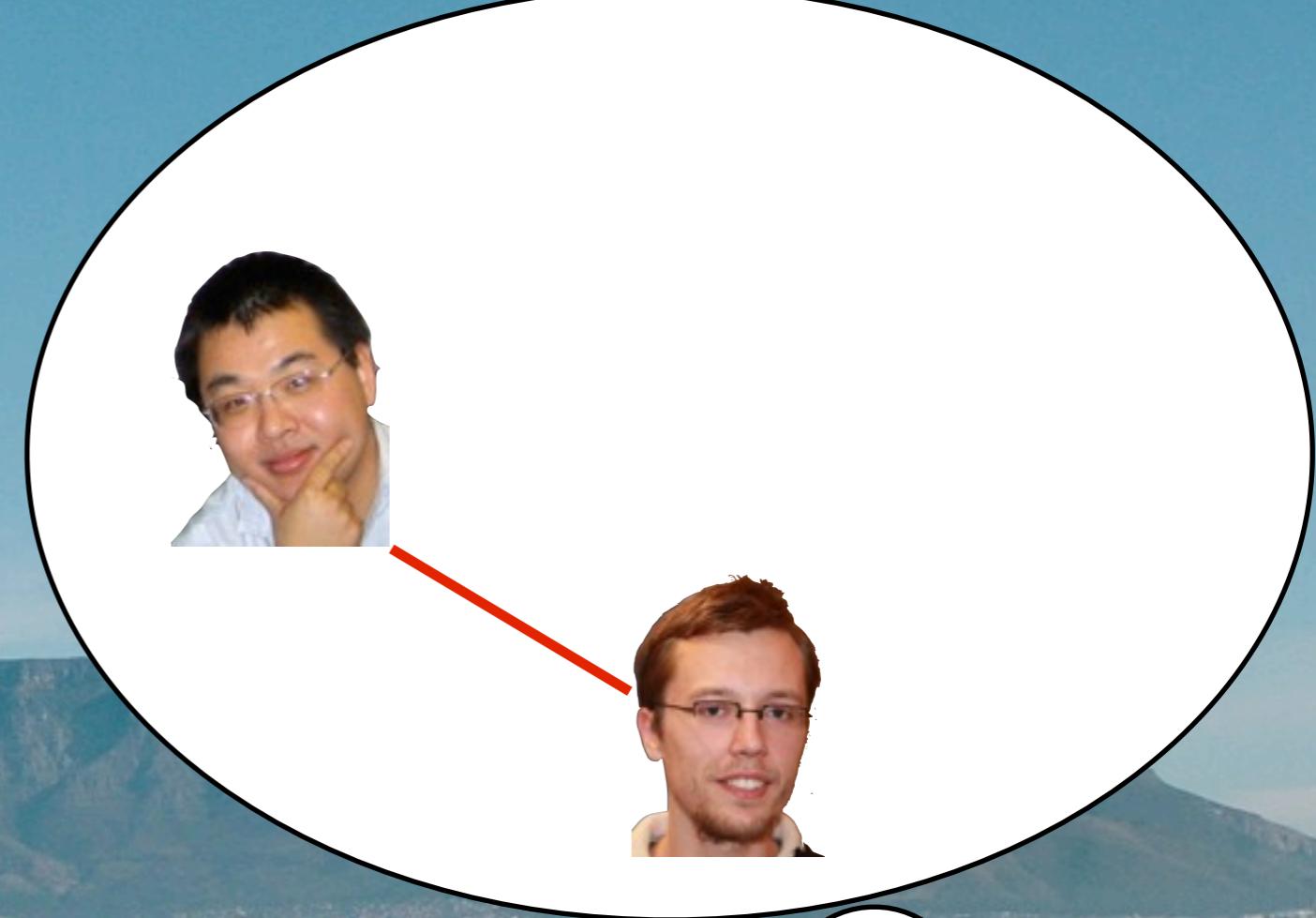


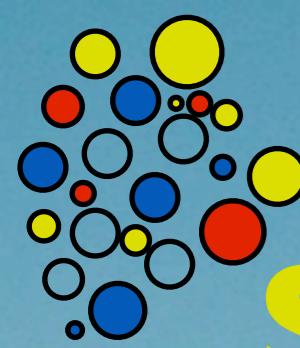
## S2. Mutual Information



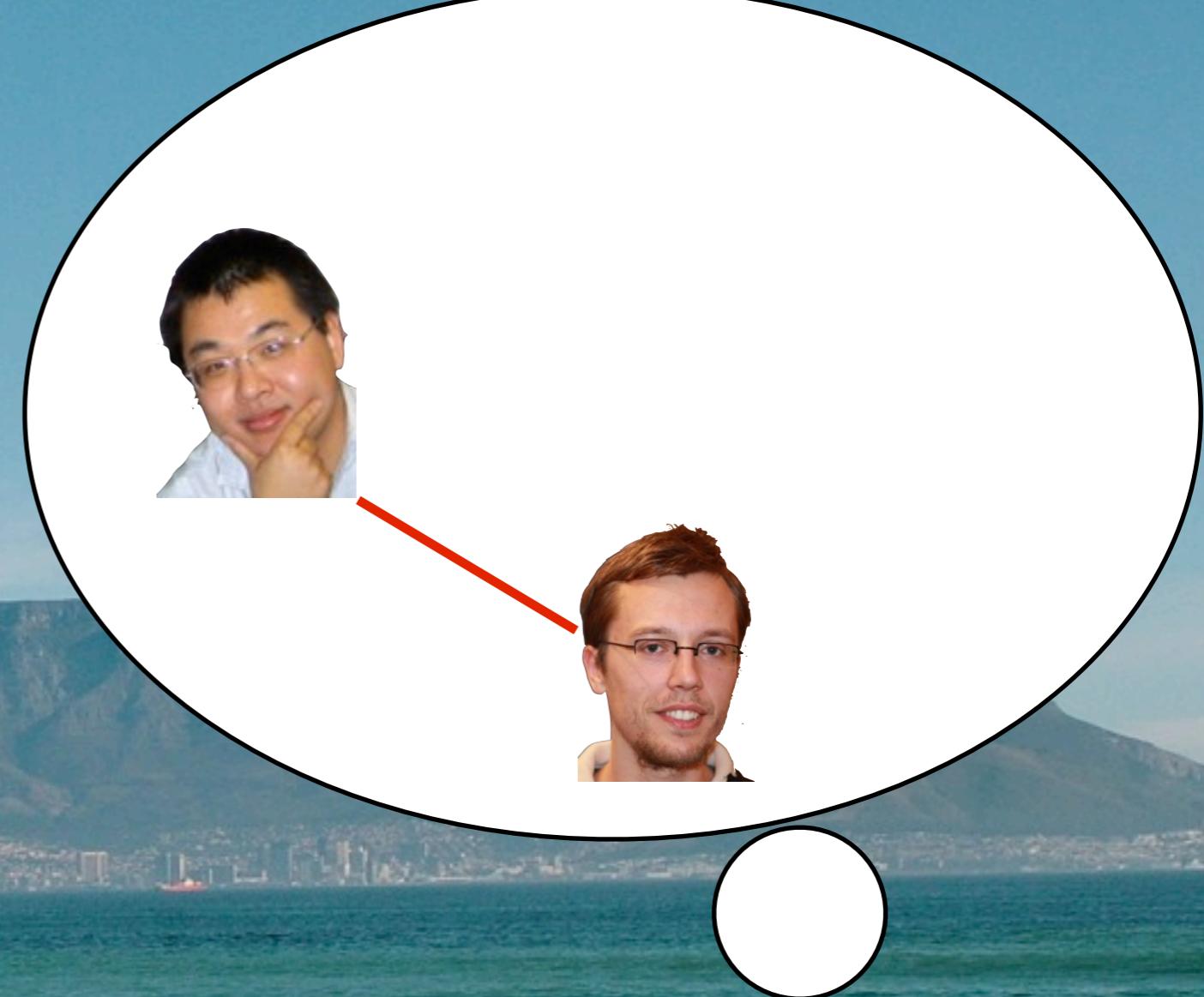


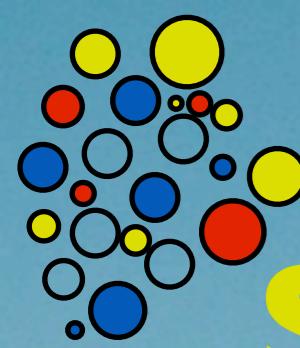
## S2. Mutual Information



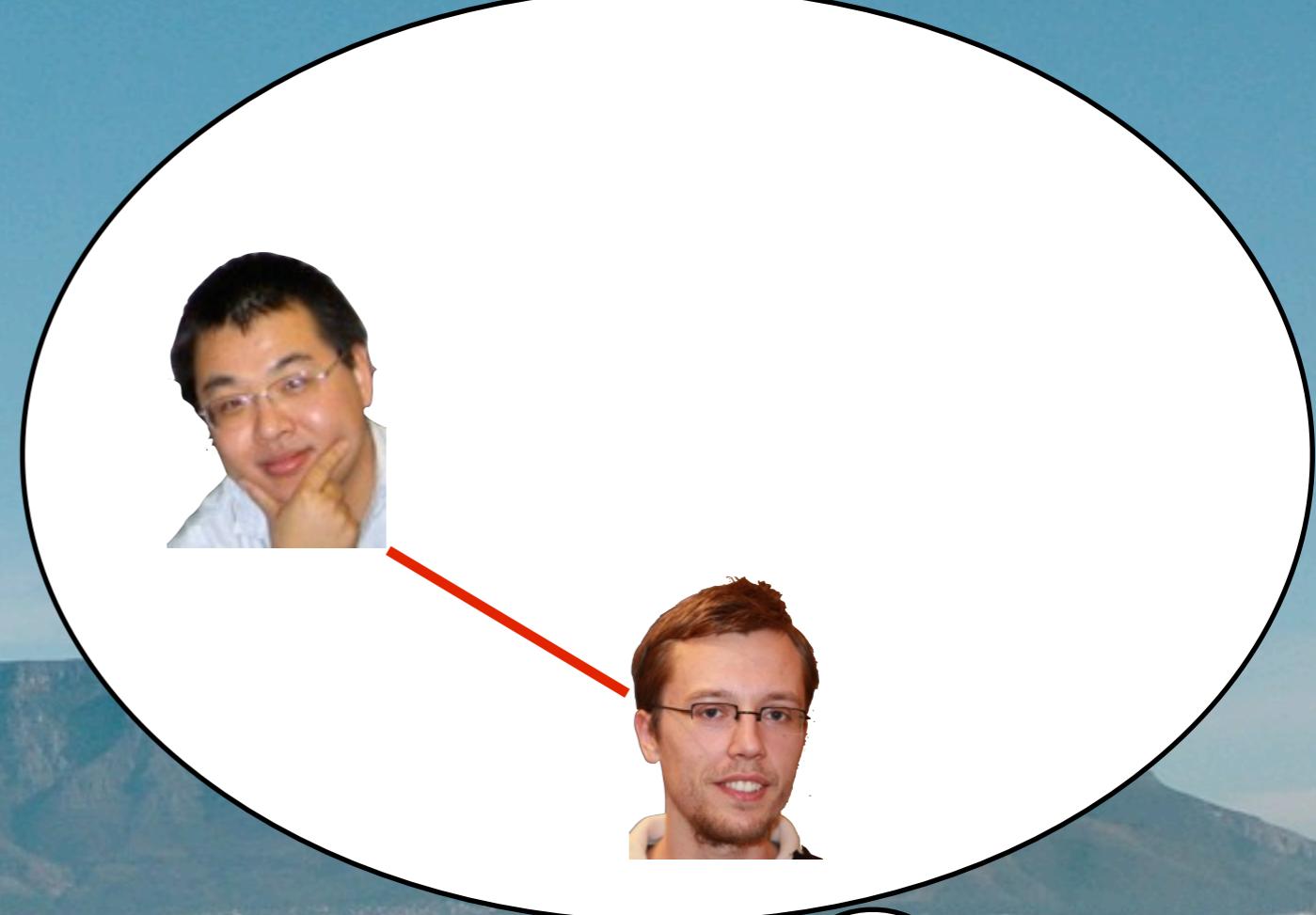


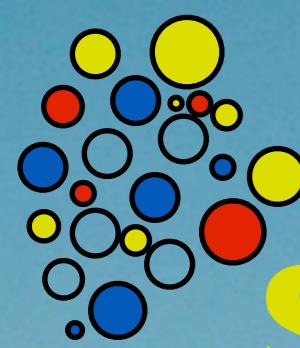
## S2. Mutual Information



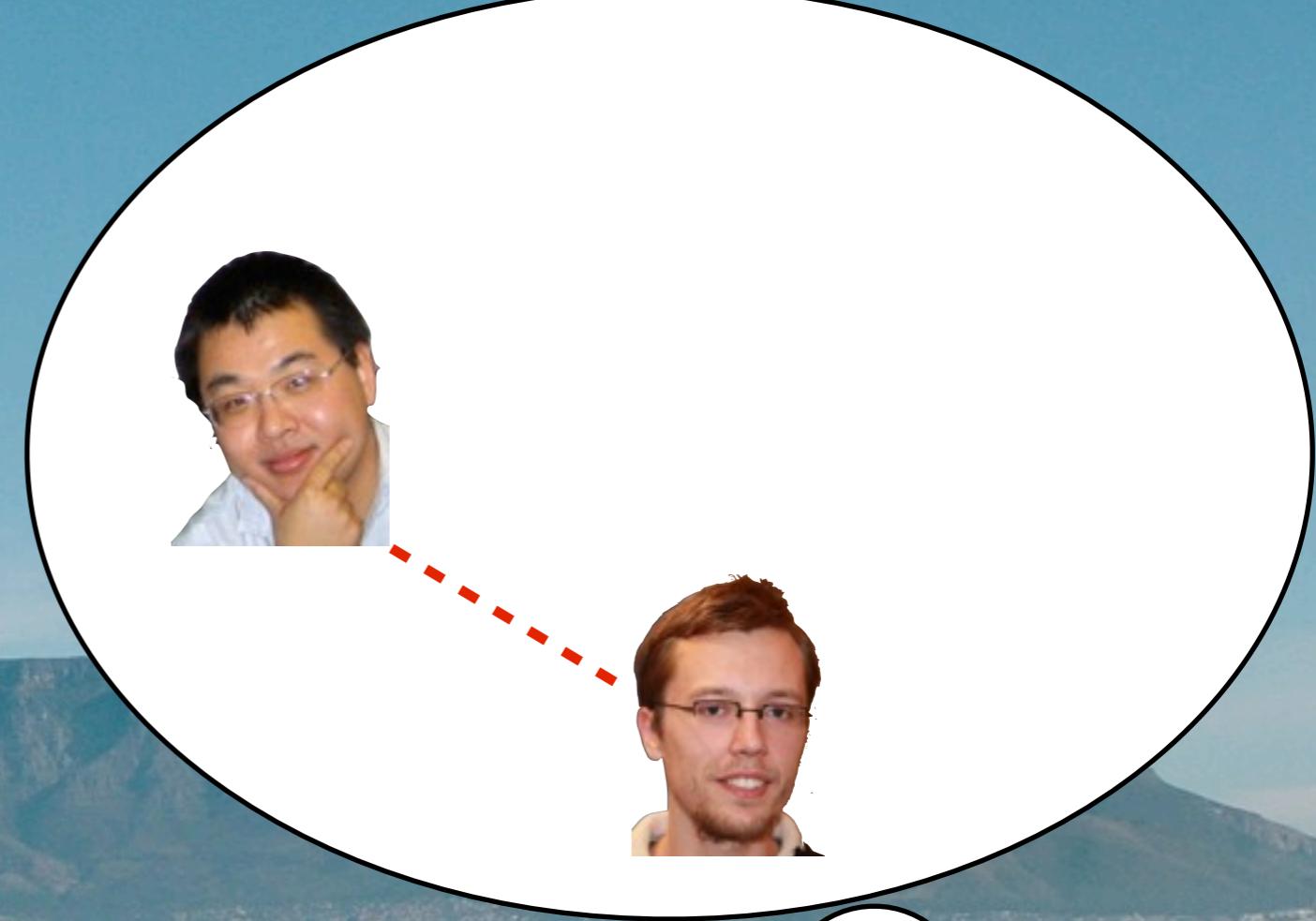


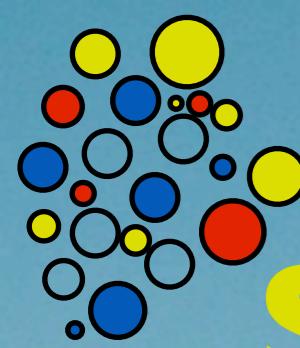
## S2. Mutual Information



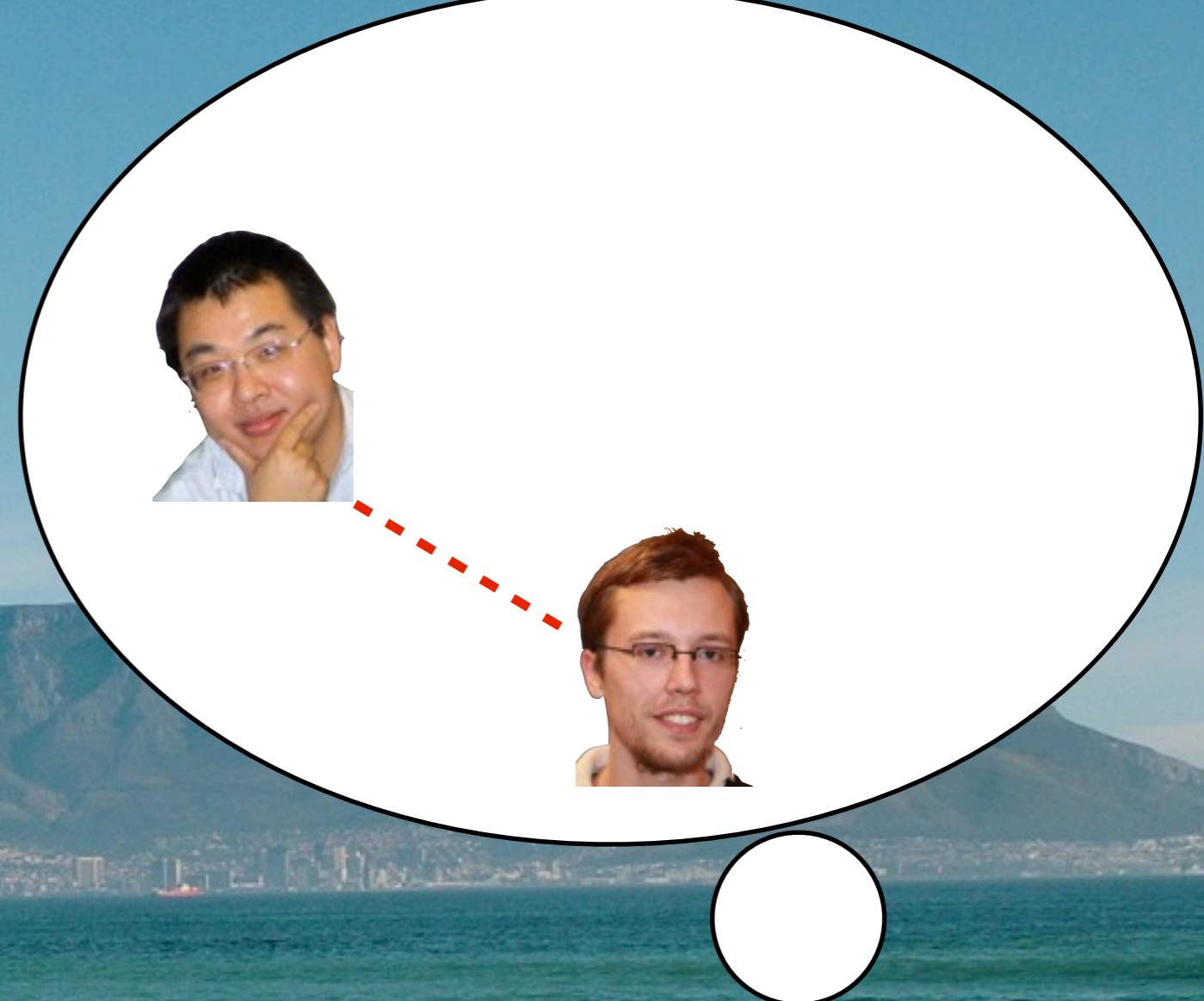
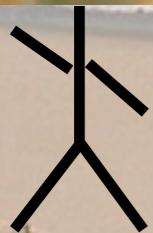


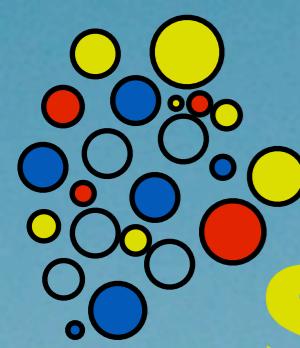
## S2. Mutual Information



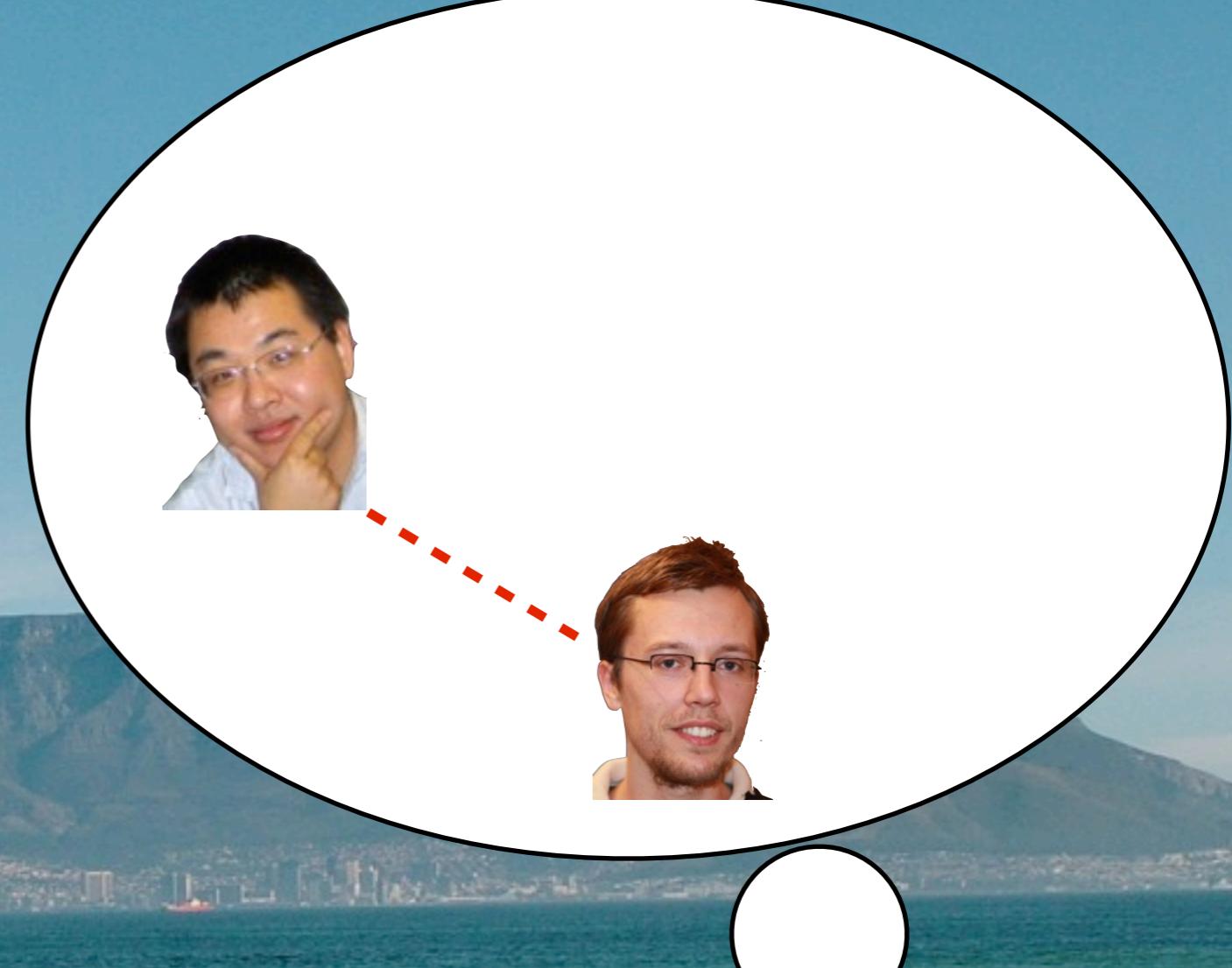
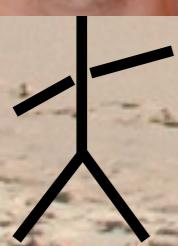


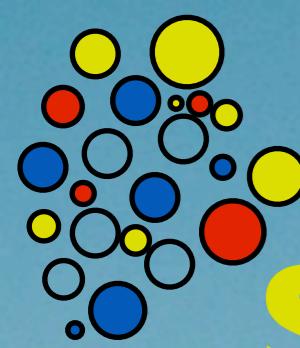
## S2. Mutual Information



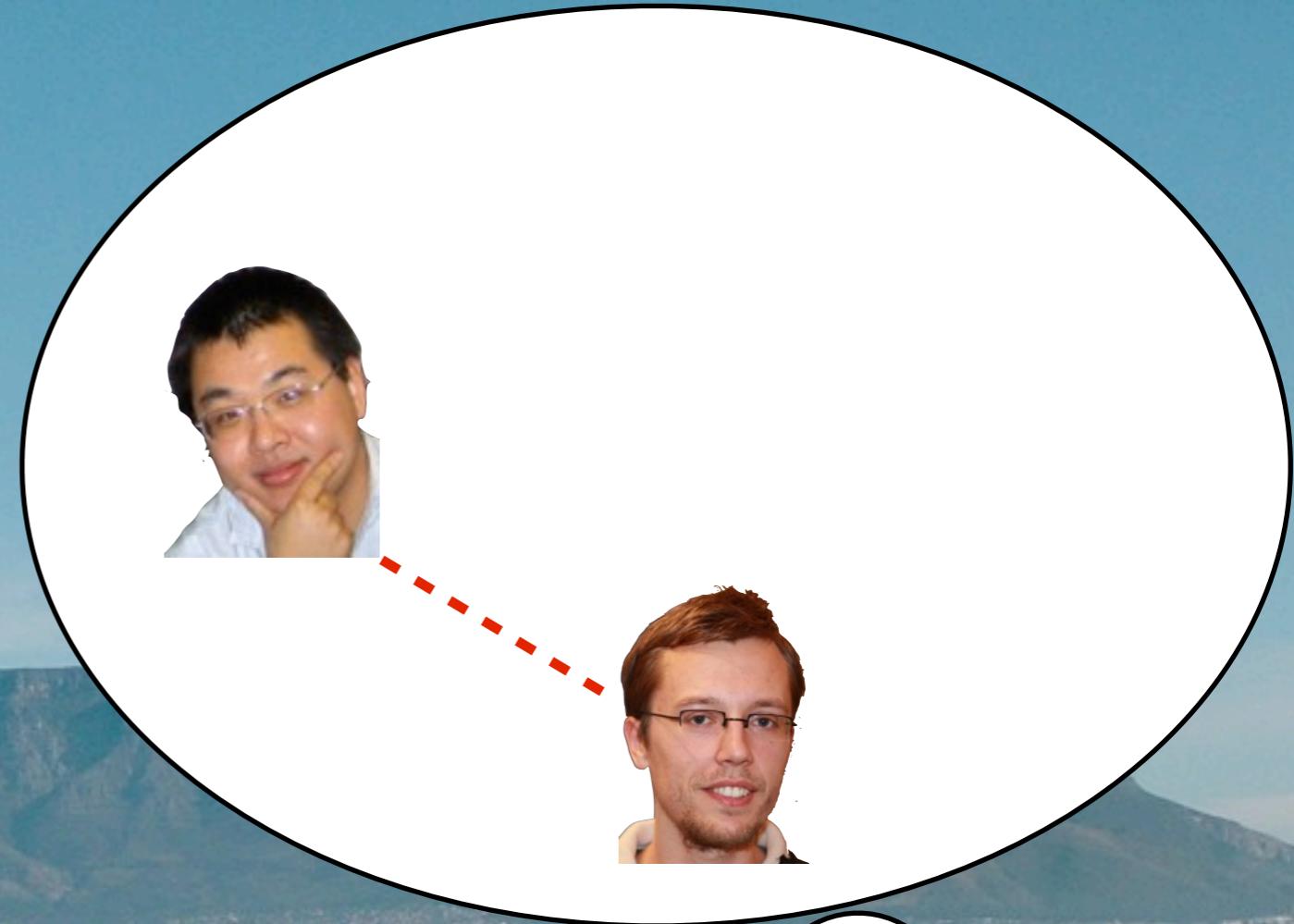
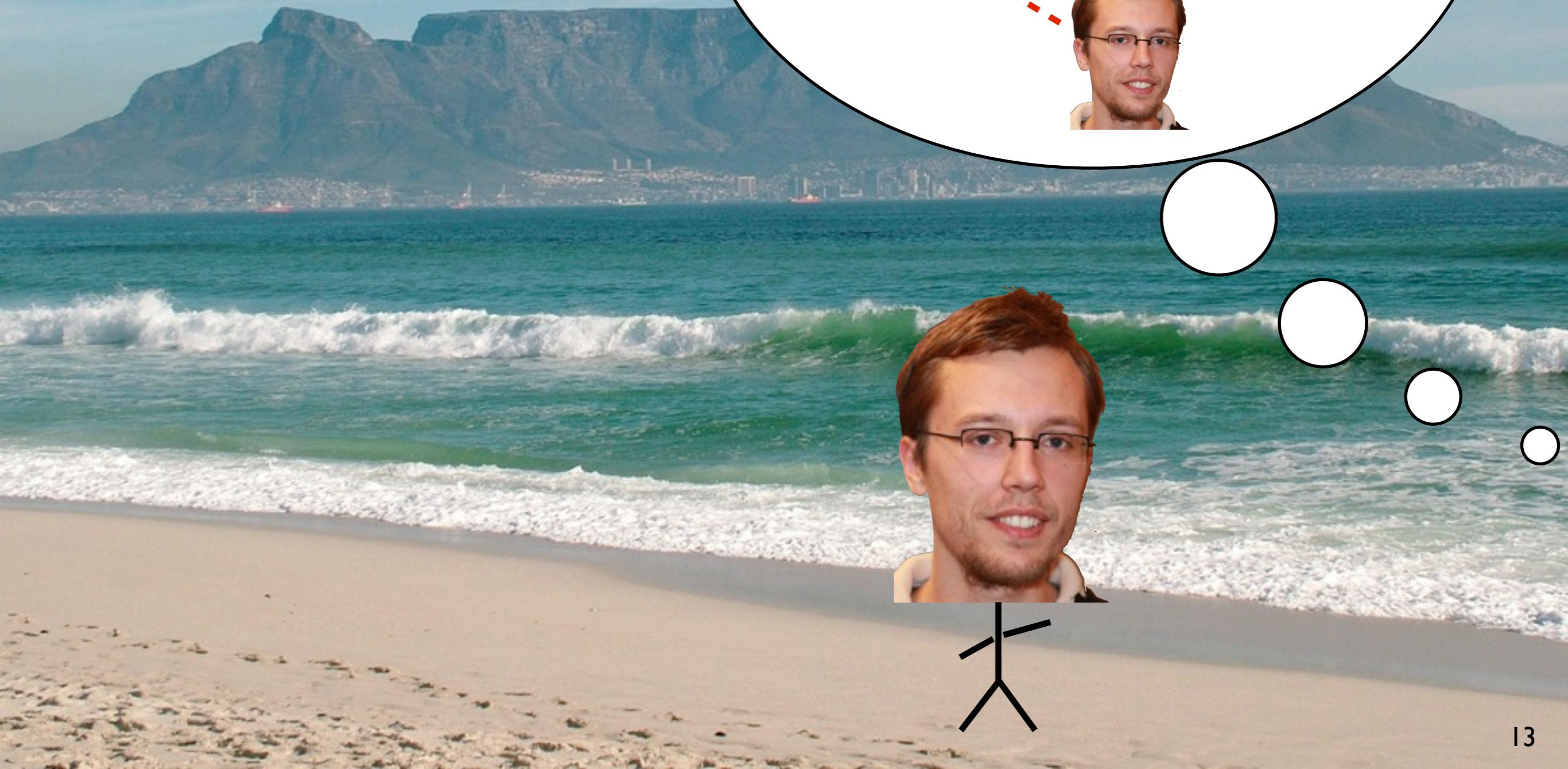


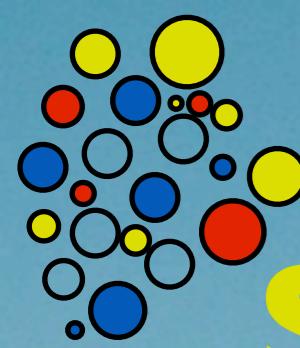
## S2. Mutual Information



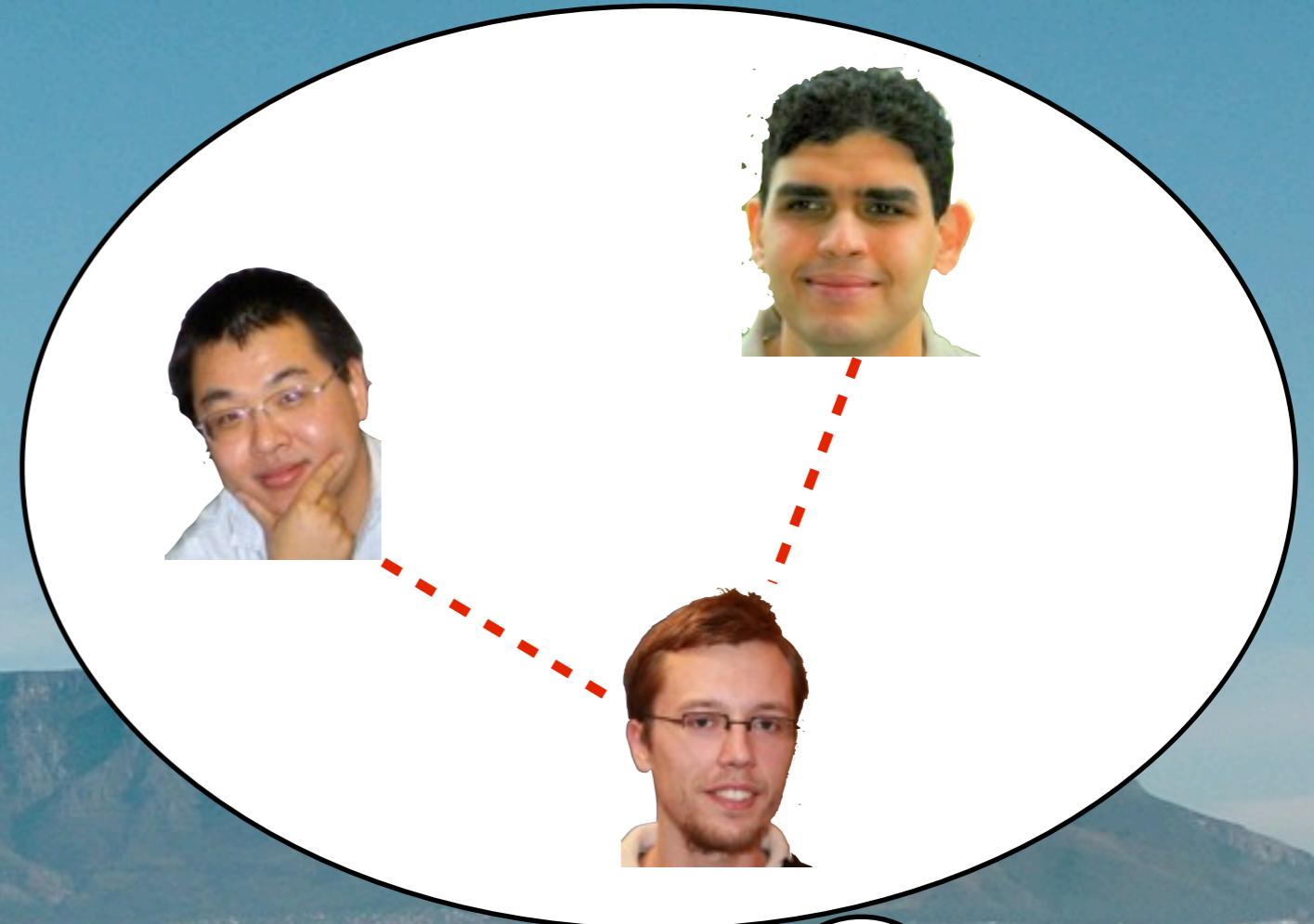


## S2. Mutual Information



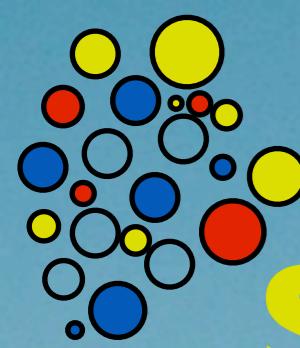


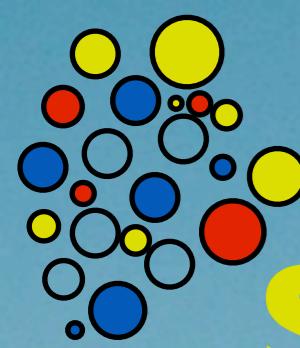
## S2. Mutual Information



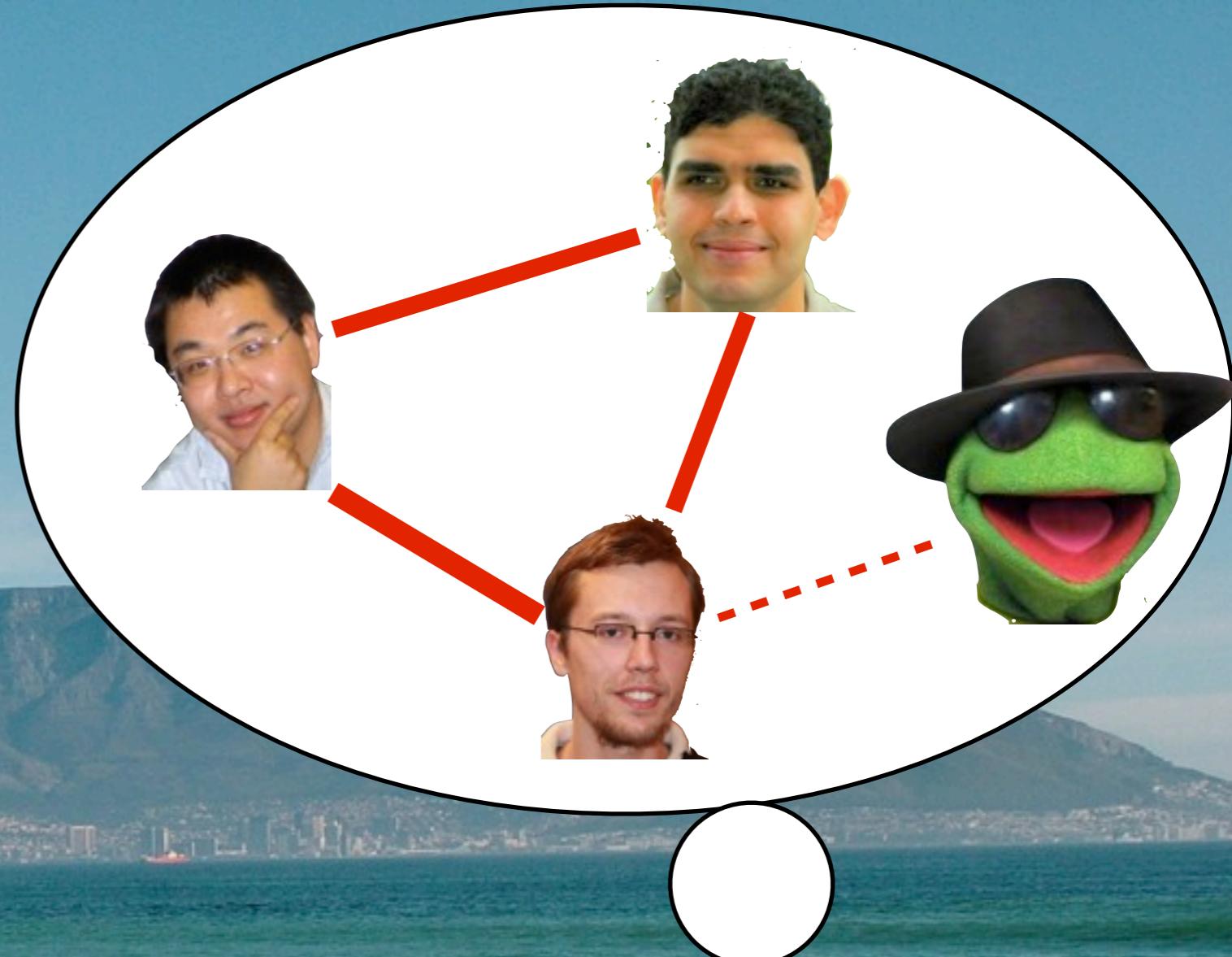


## S2. Mutual Information



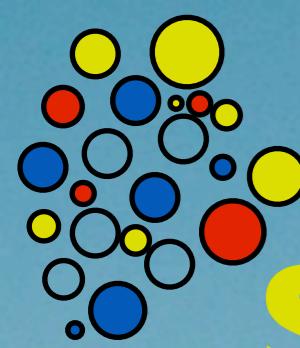


## S2. Mutual Information

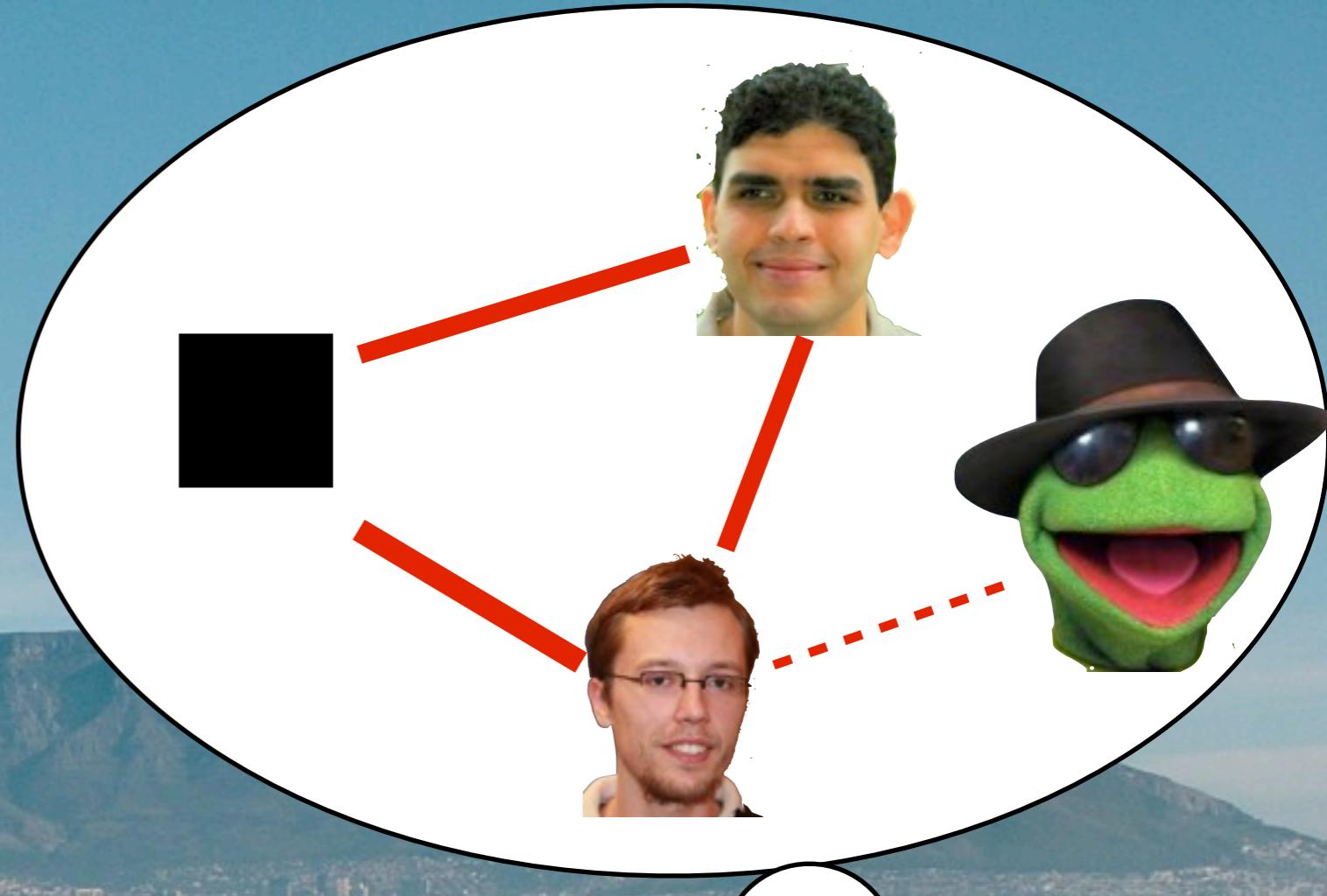


How much does occurrence of  
reveal about occurrence of ?





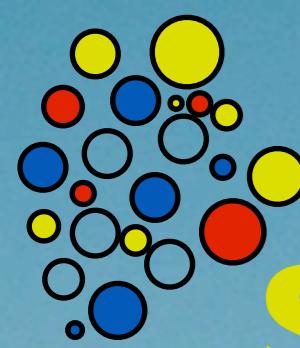
## S2. Mutual Information



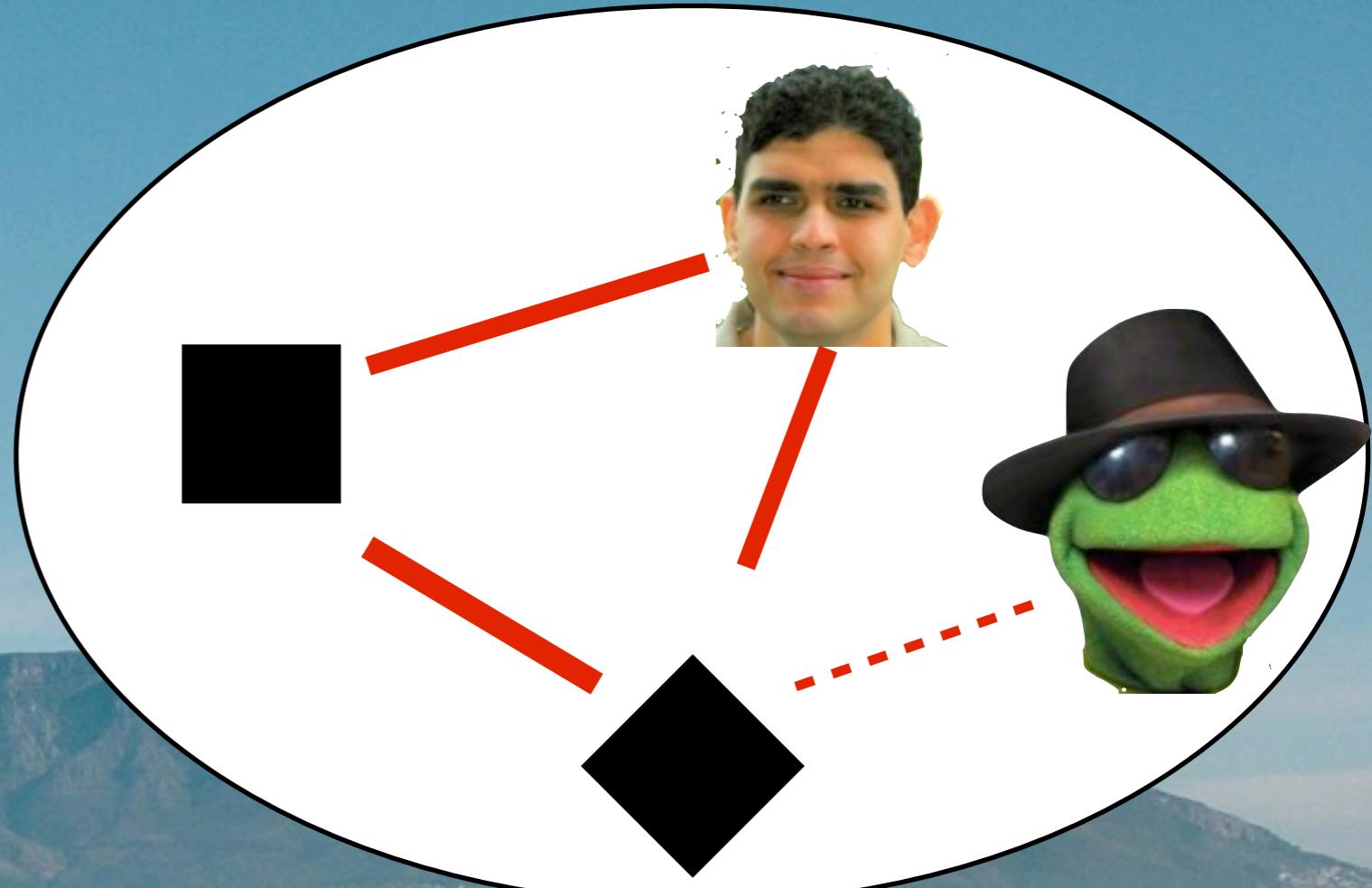
How much does occurrence of  
reveal about occurrence of



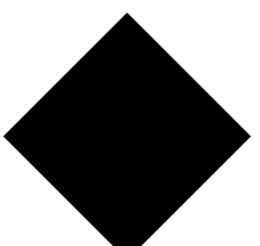
?

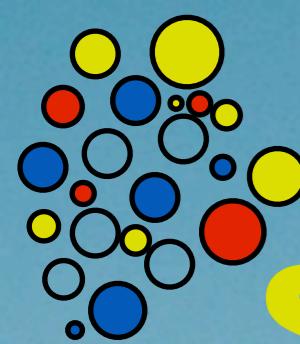


## S2. Mutual Information

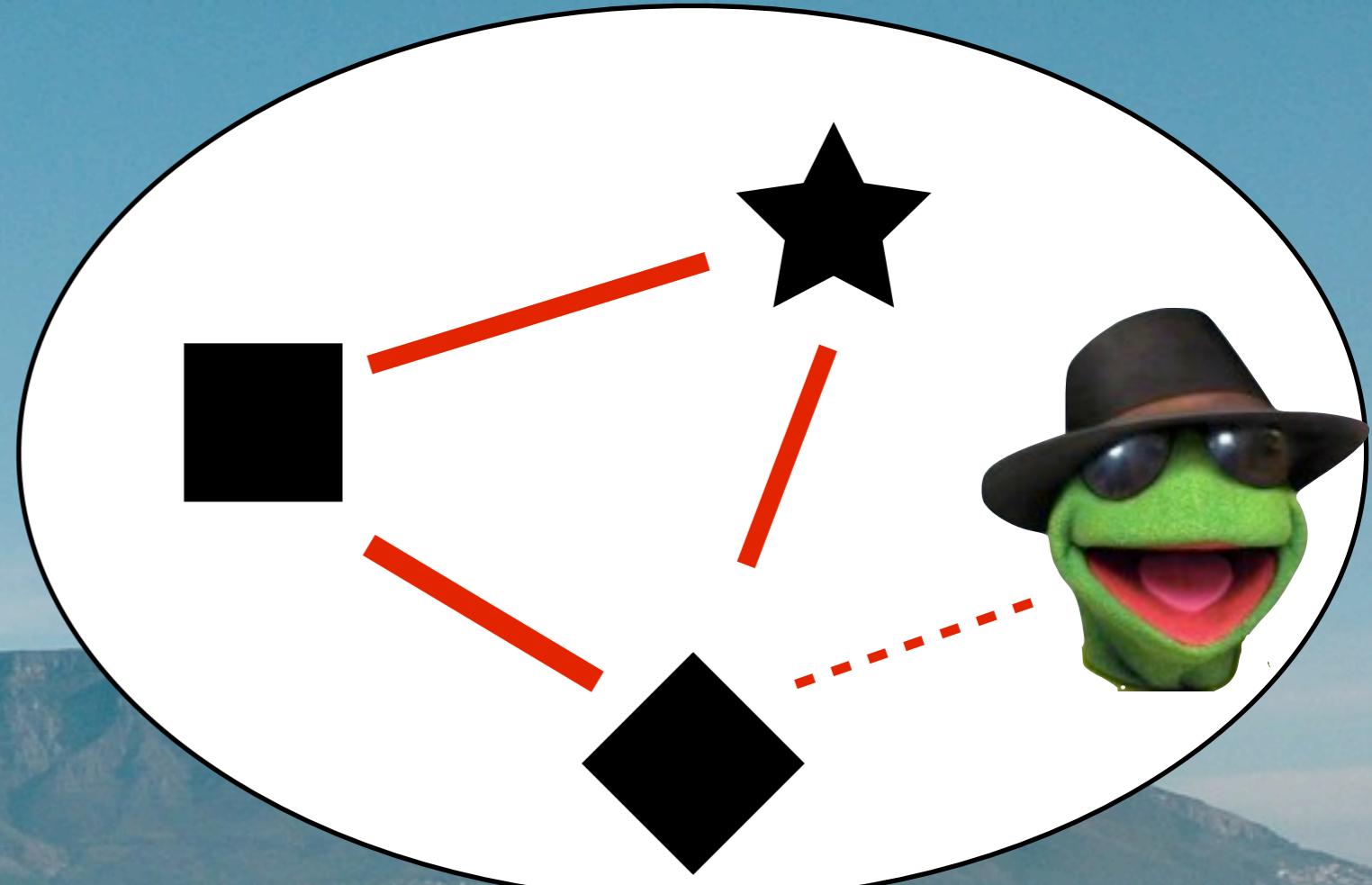


How much does occurrence of  
reveal about occurrence of ?

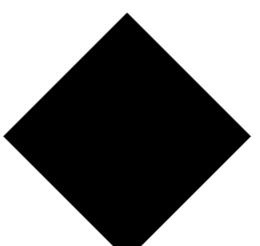


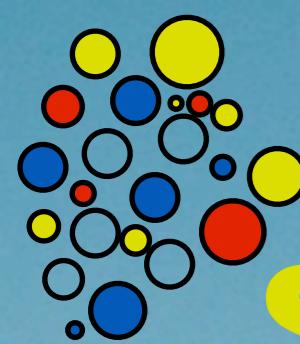


## S2. Mutual Information

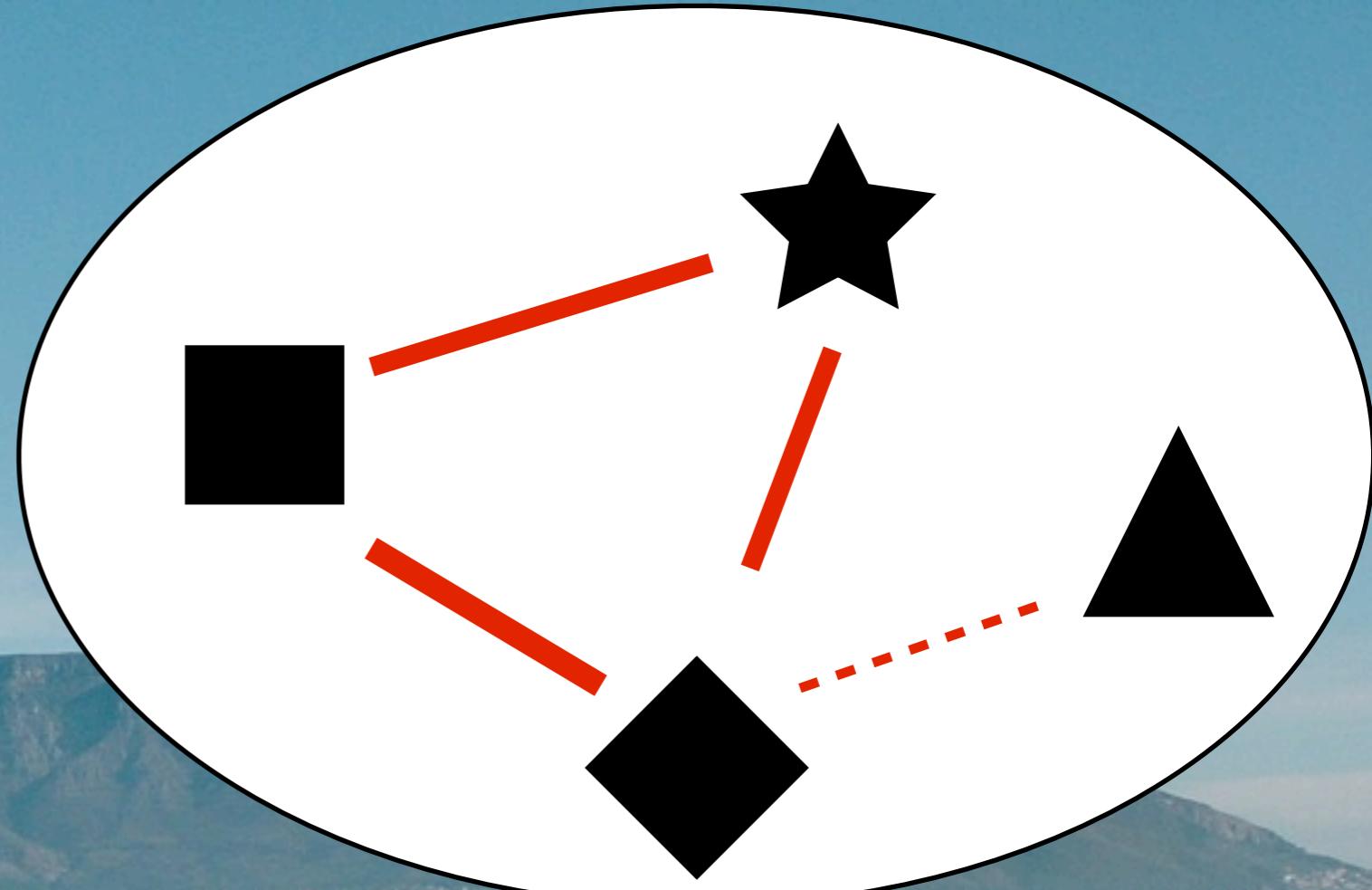


How much does occurrence of  
reveal about occurrence of ?

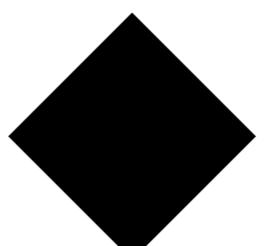




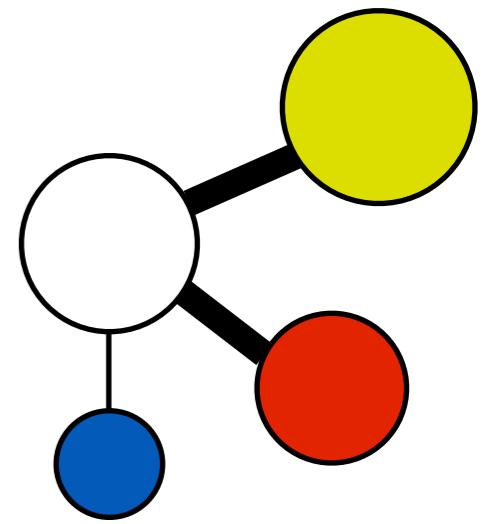
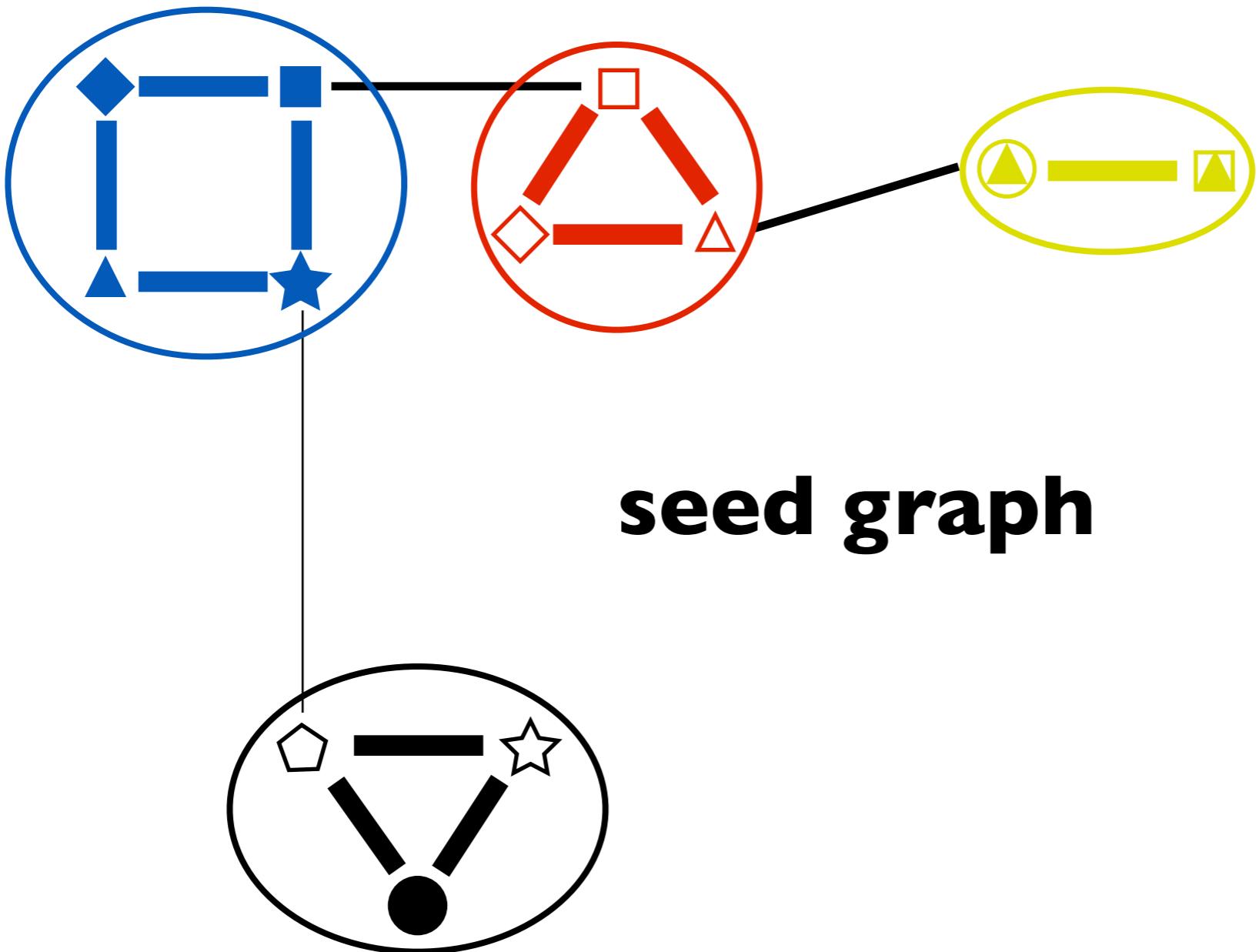
## S2. Mutual Information



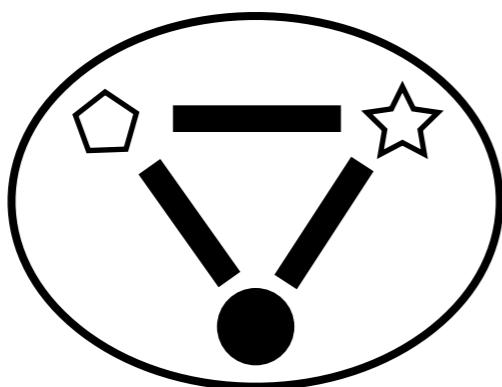
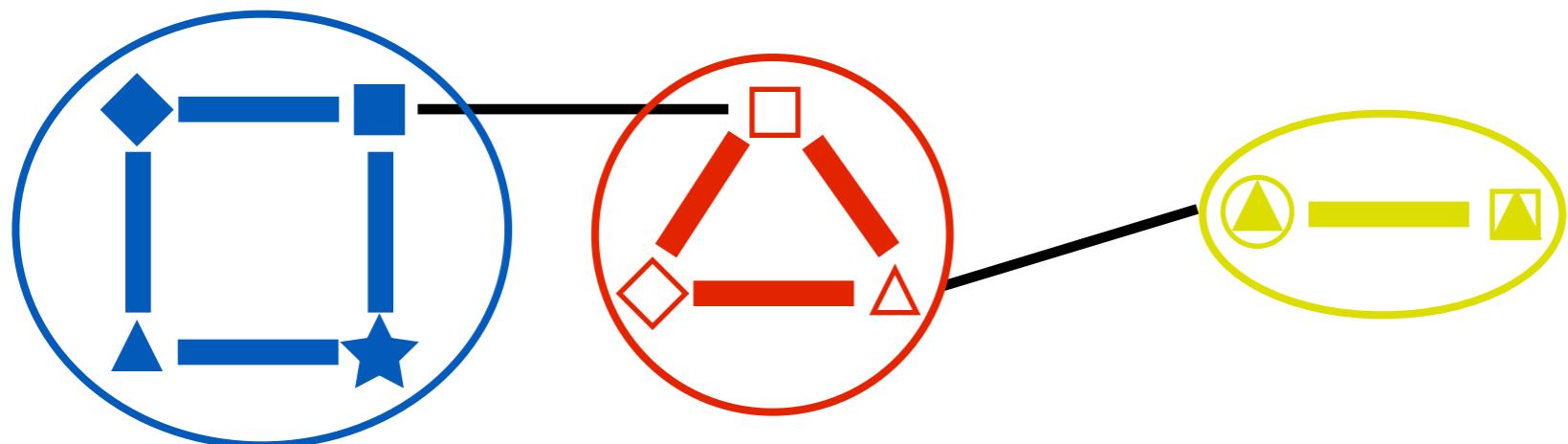
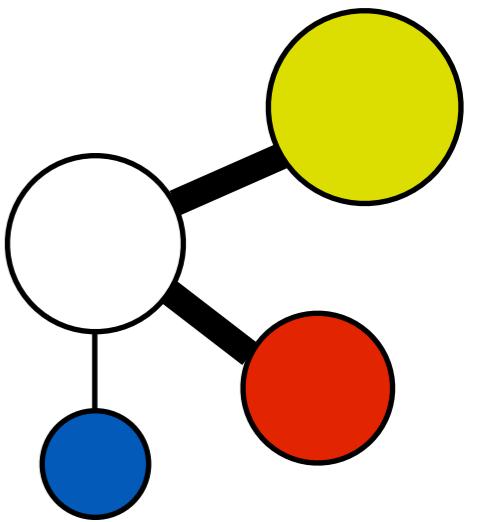
How much does occurrence of  
reveal about occurrence of ?



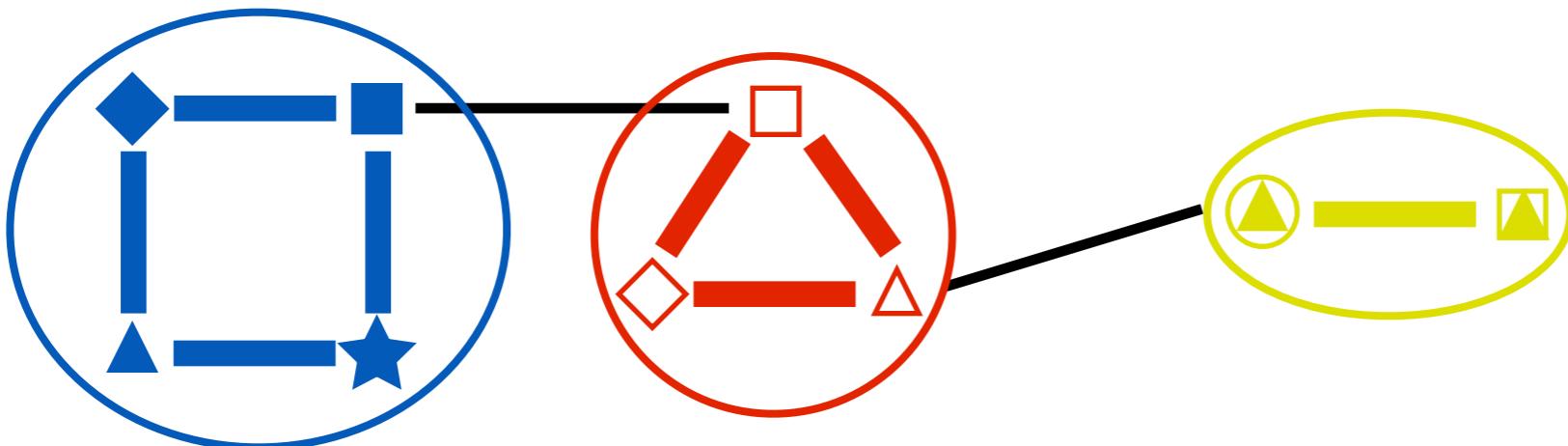
# S3. Concern Relations



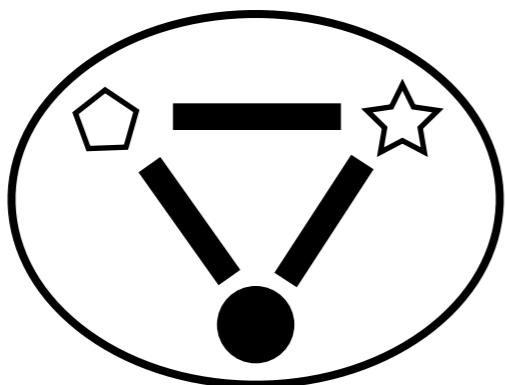
# S3. Concern Relations



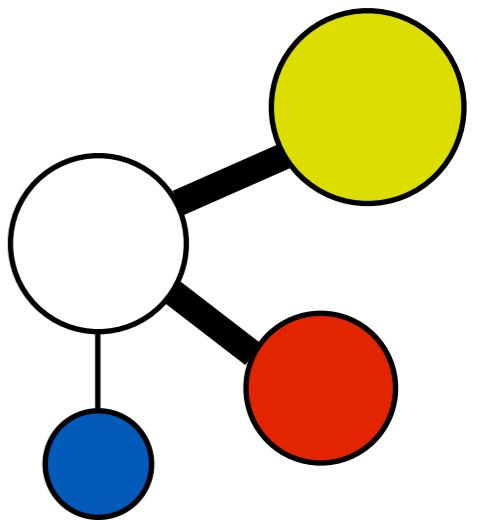
# S3. Concern Relations



**composite**  
concern



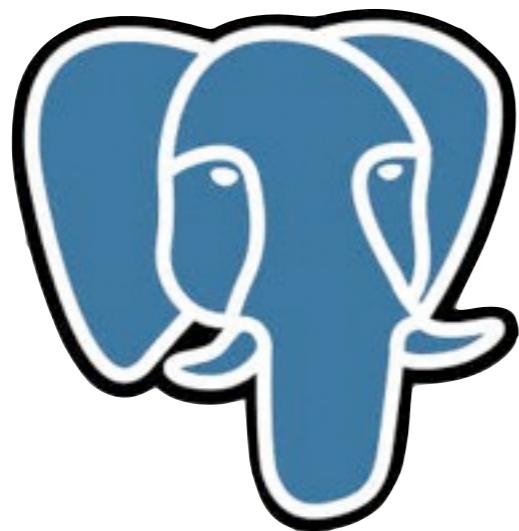
**simple**  
concern



- 1. What is a Crosscutting Concern?**
- 2. The Concern Mining Process and its Shortcomings**
- 3. COMMIT**
- 4. Case Study**
- 5. Conclusion**

# Case Study

PostgreSQL



1996-2002  
(800 kLOC)



1993-2003  
(2 MLOC)

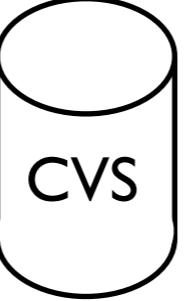
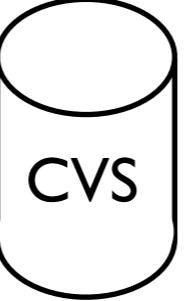
# Comparative Study

limited context	noise	no composition

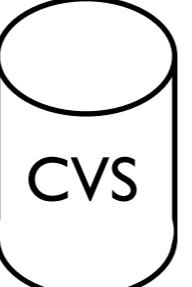
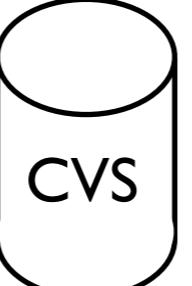
# Comparative Study

	limited context	noise	no composition
CBFA			
HAM			
COMMIT			

# Comparative Study

	limited context	noise	no composition
CBFA	snapshot		
HAM	 CVS		
COMMIT	 CVS		

# Comparative Study

	limited context	noise	no composition
CBFA	snapshot	<b>similar</b> entity names	
HAM	 CVS	<b>identical</b> set of callers	
COMMIT	 CVS	mutual information	

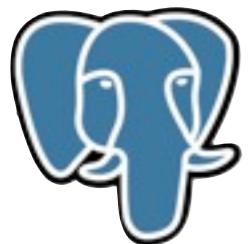
# Comparative Study

	limited context	noise	no composition
CBFA	snapshot	<b>similar</b> entity names	
HAM		<b>identical</b> set of callers	
COMMIT		mutual information	

# Study Design

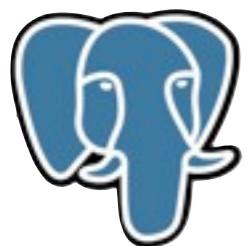
# Study Design

PostgreSQL



# Study Design

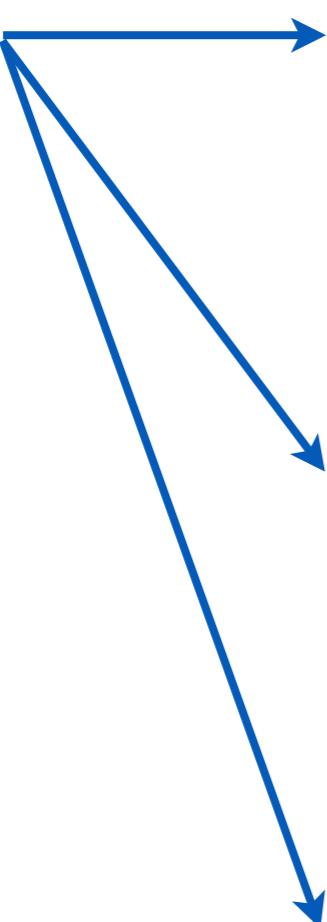
PostgreSQL



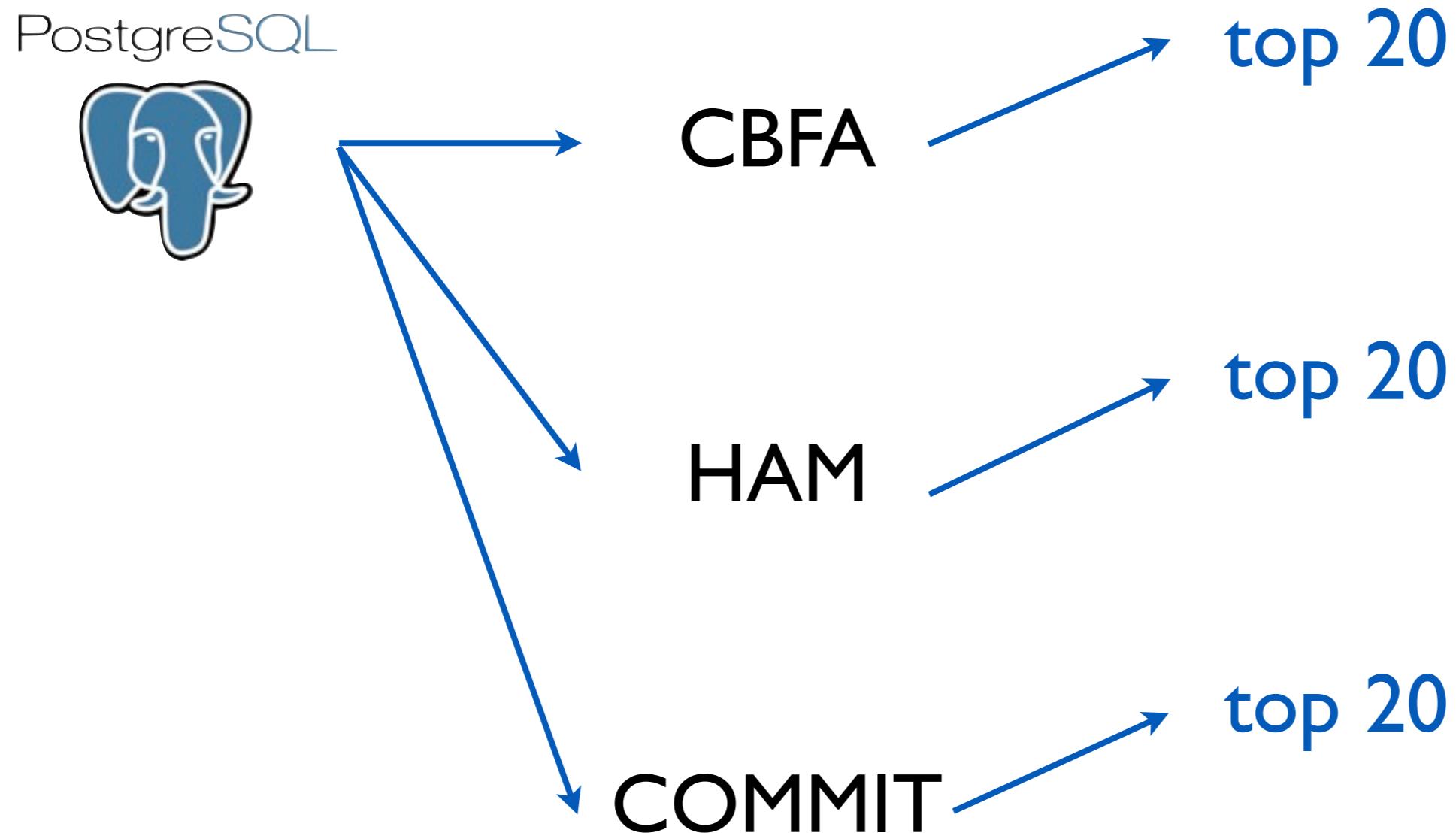
**CBFA**

**HAM**

**COMMIT**

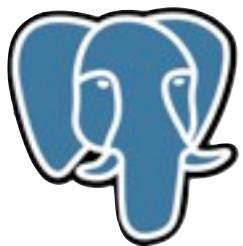


# Study Design



# Study Design

PostgreSQL



CBFA

HAM

COMMIT

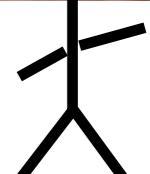
top 20

top 20

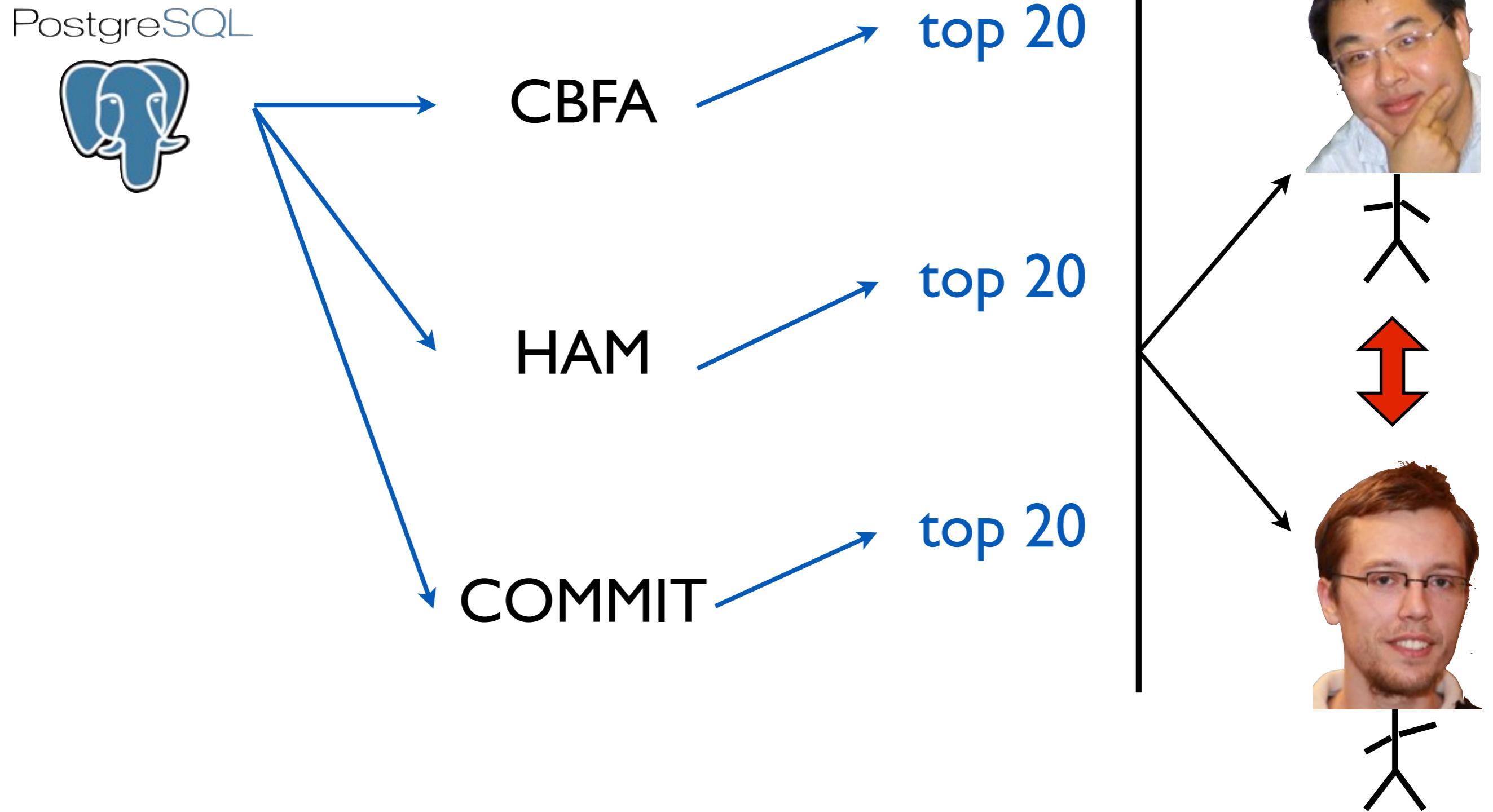
top 20



concern?



# Study Design



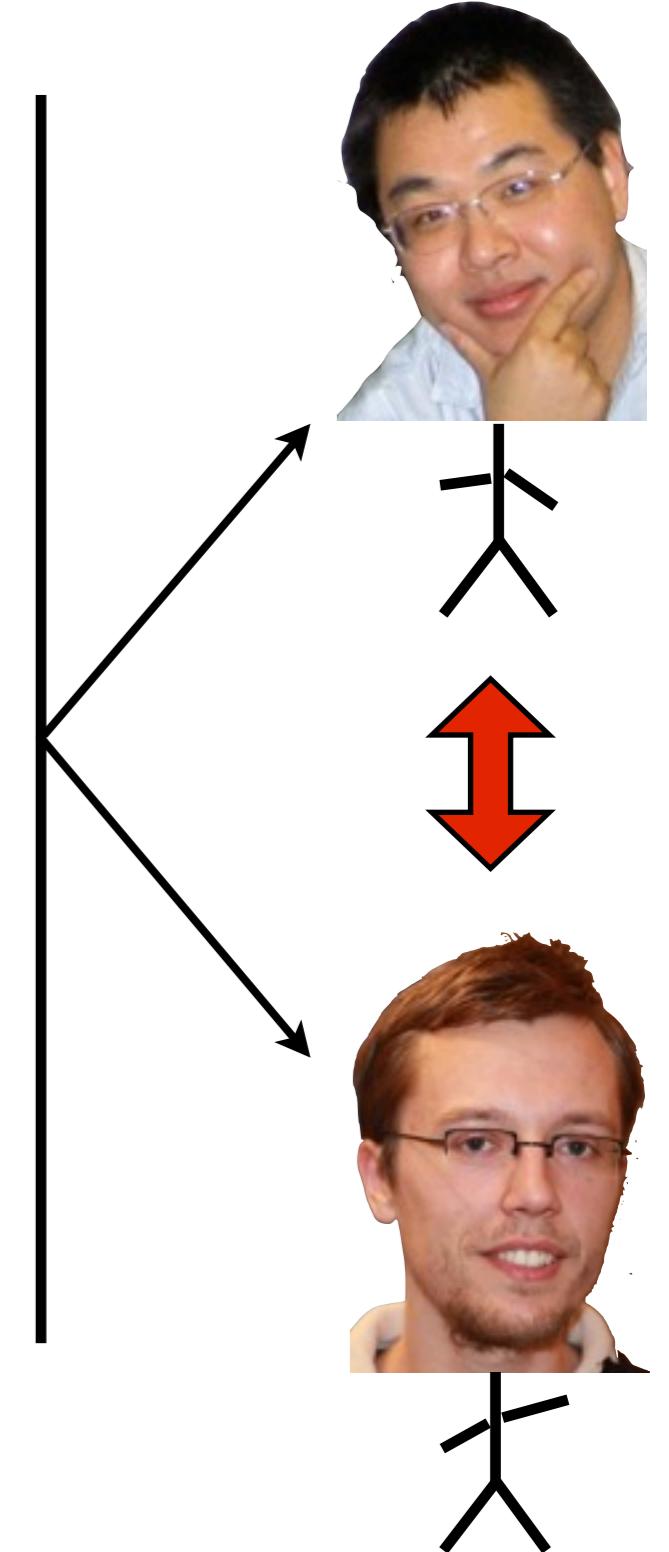
# Study Design



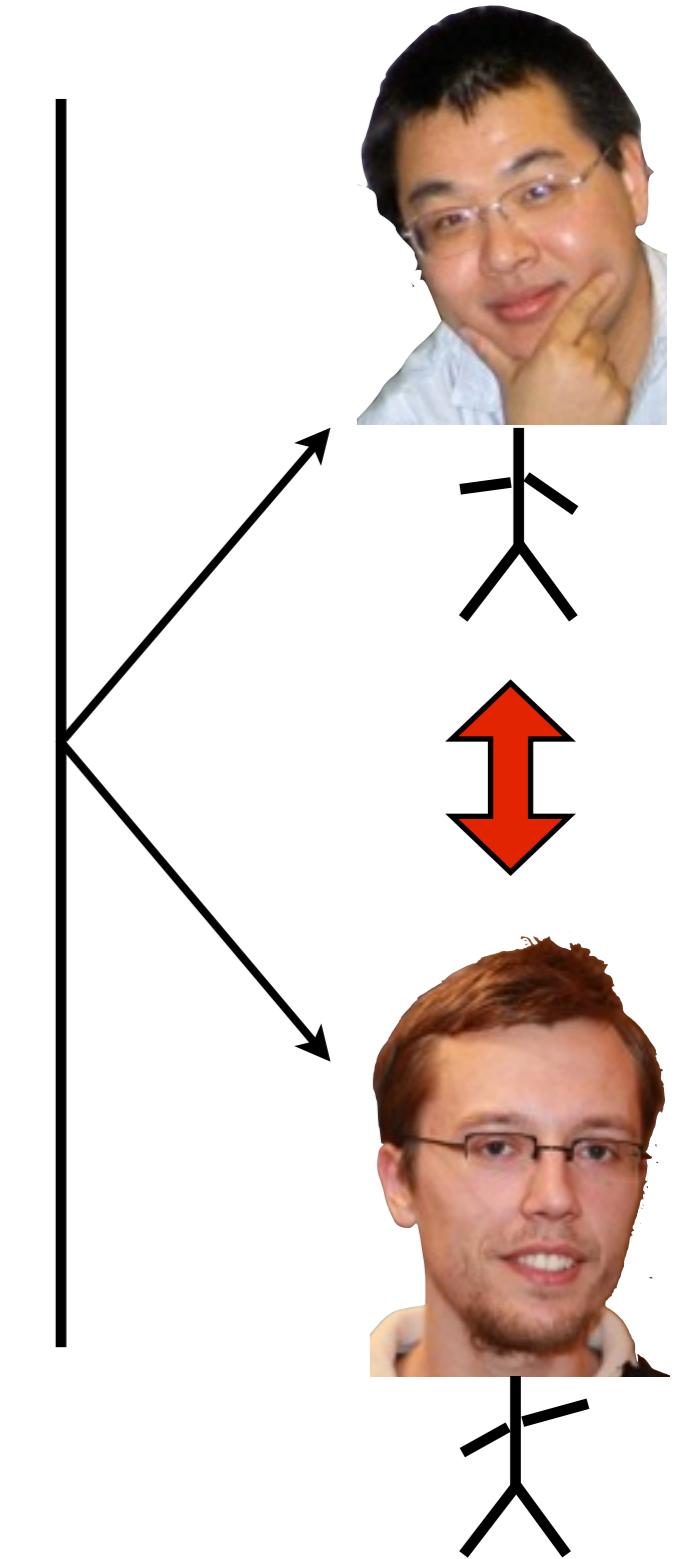
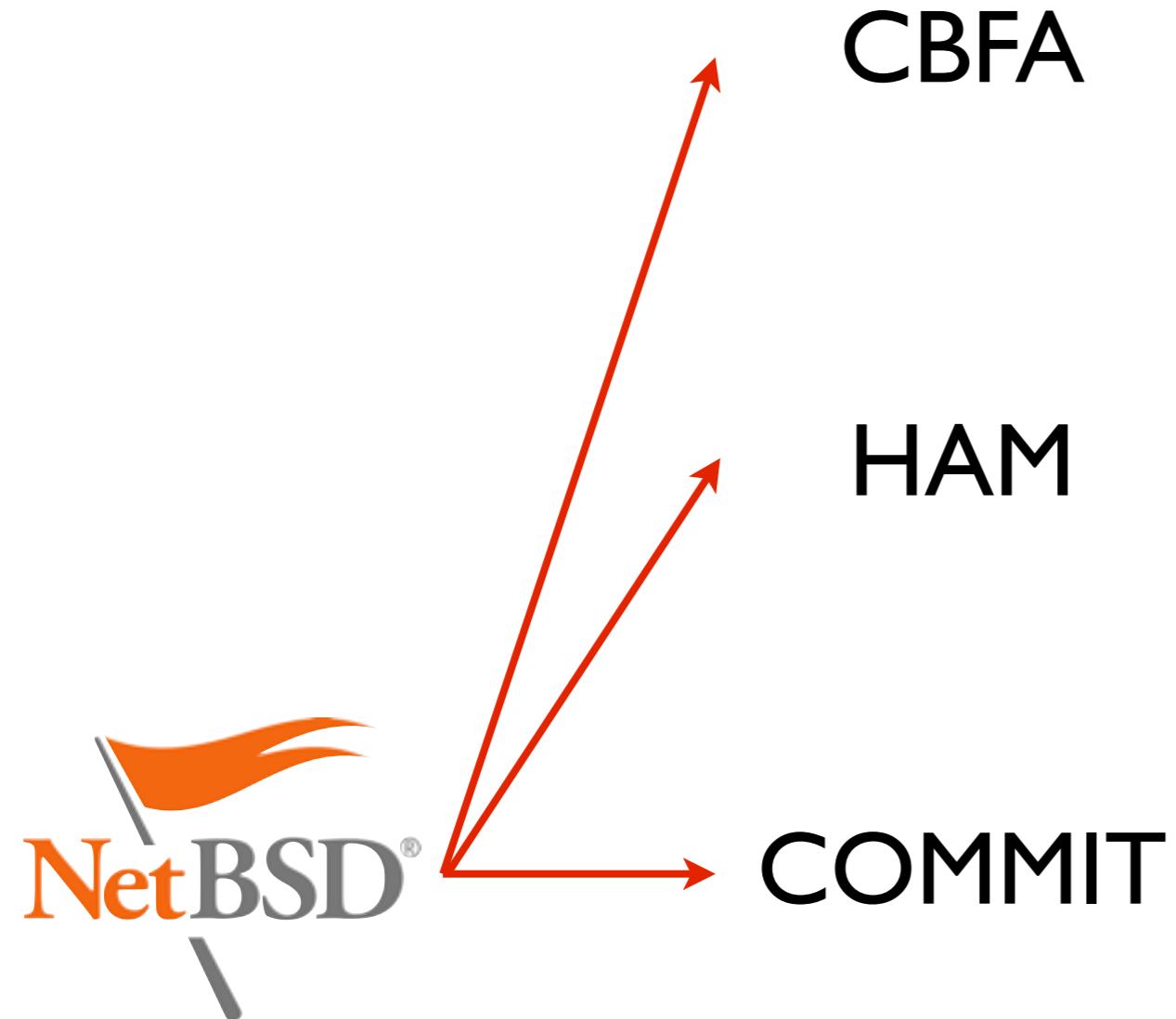
CBFA

HAM

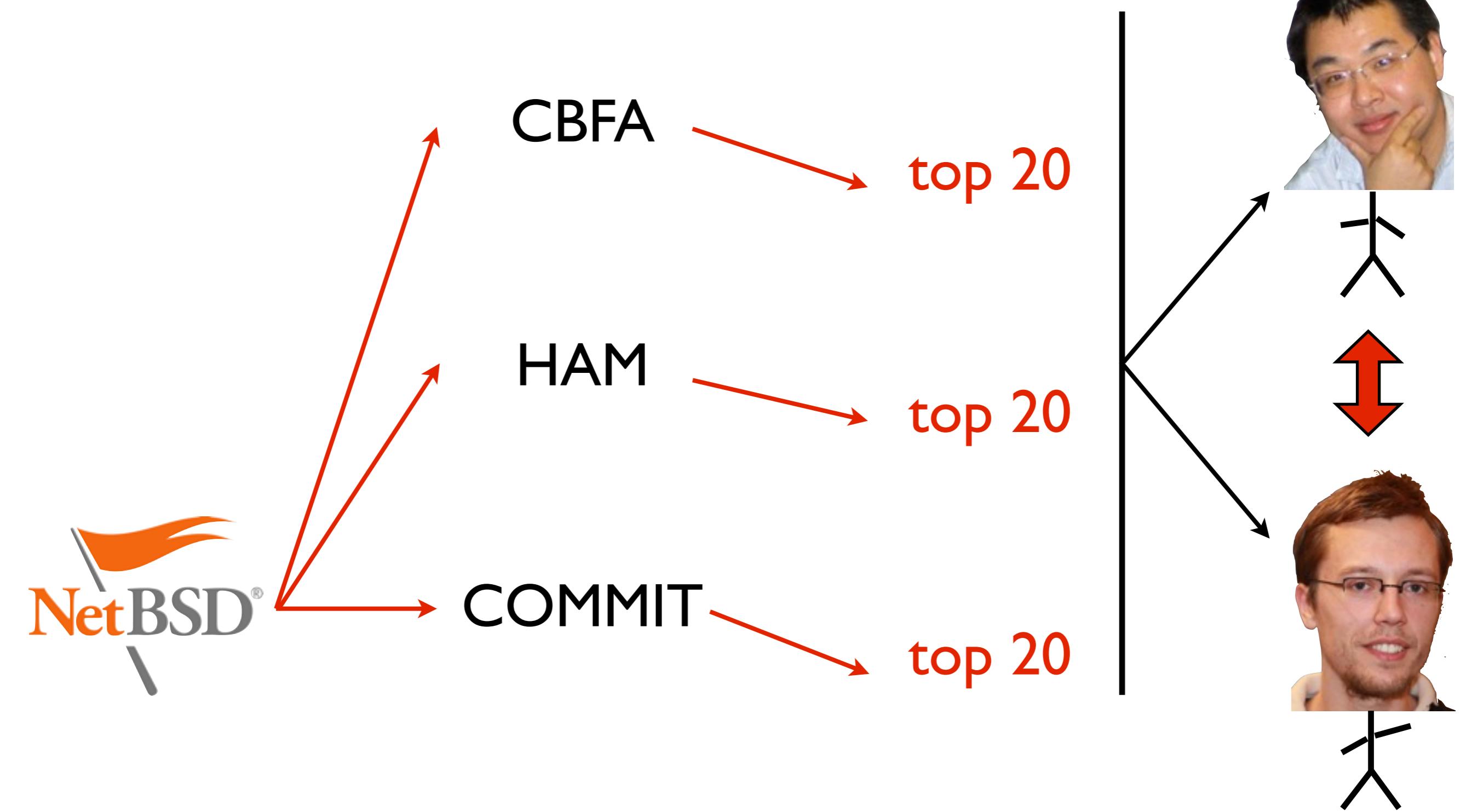
COMMIT



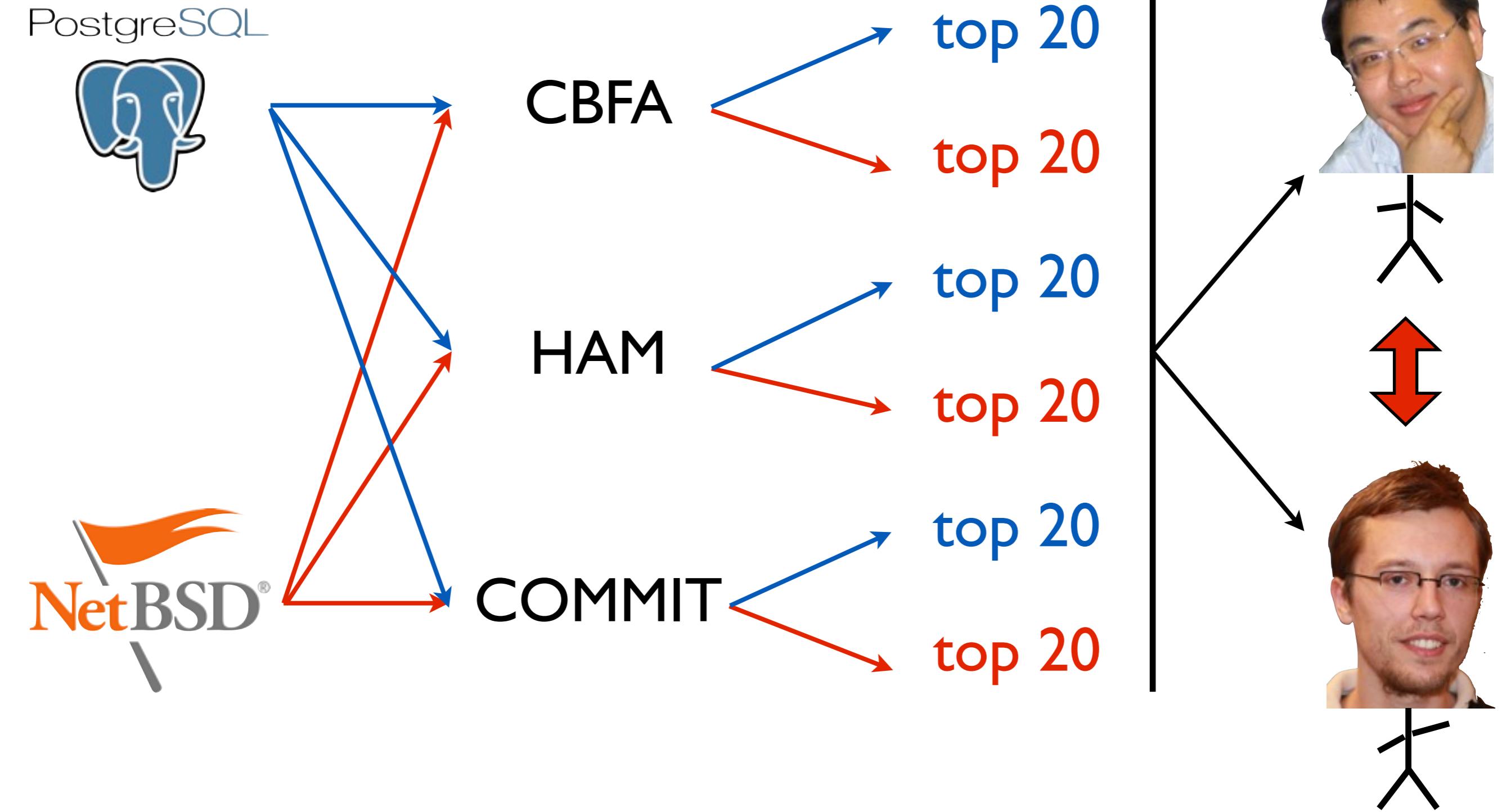
# Study Design



# Study Design



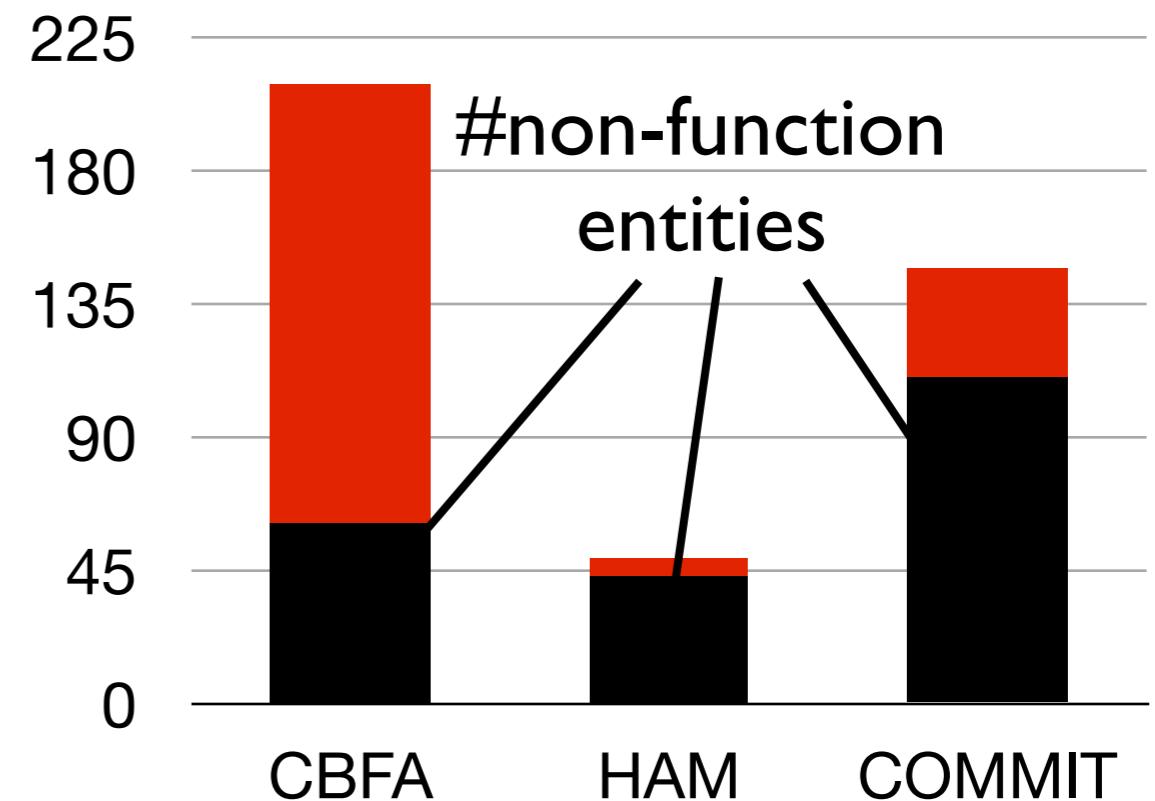
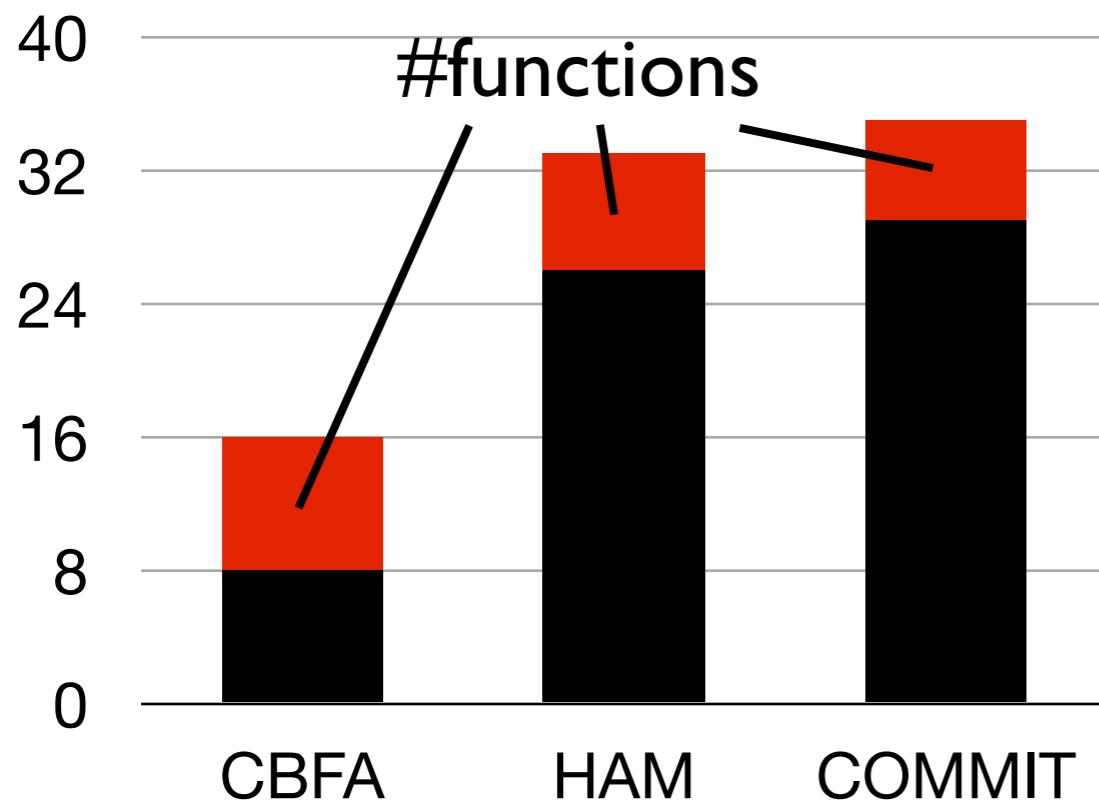
# Study Design



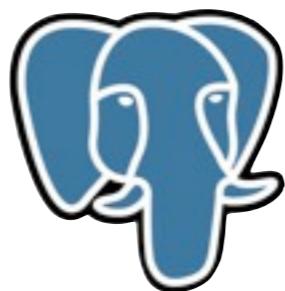
# **H I.** Richer Data Sources Yield **richer** Seeds



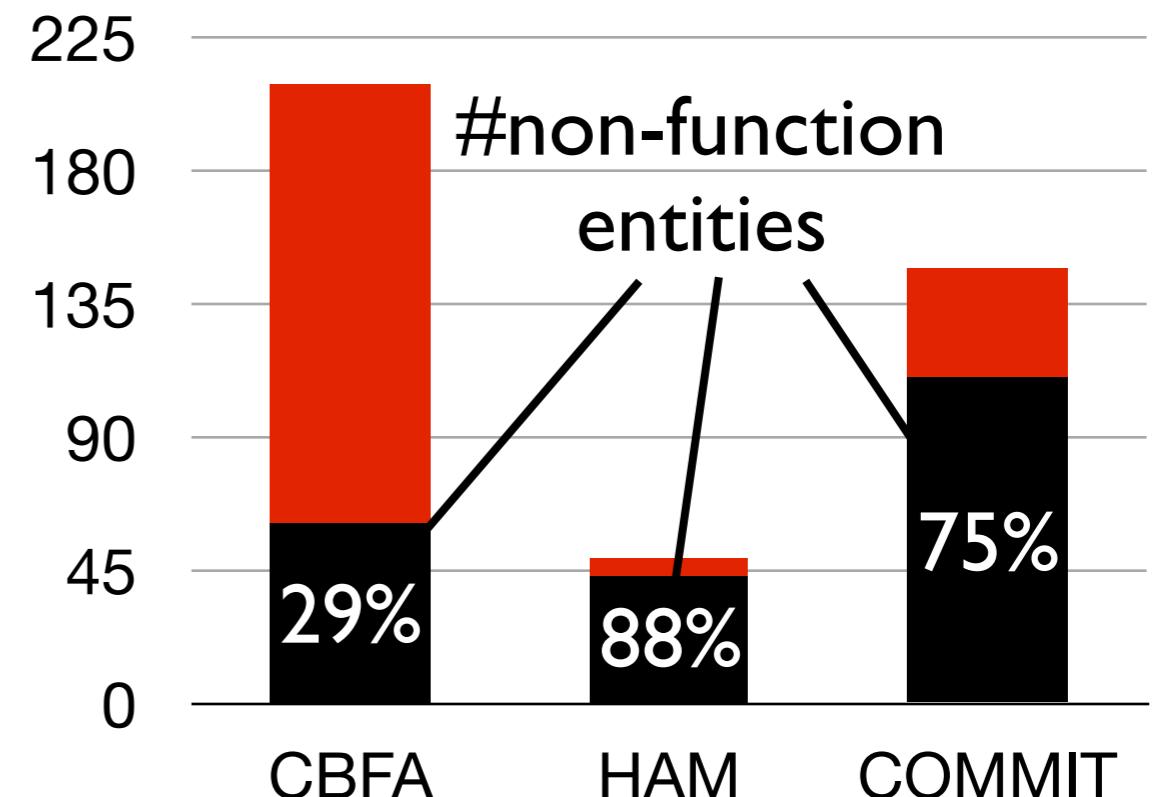
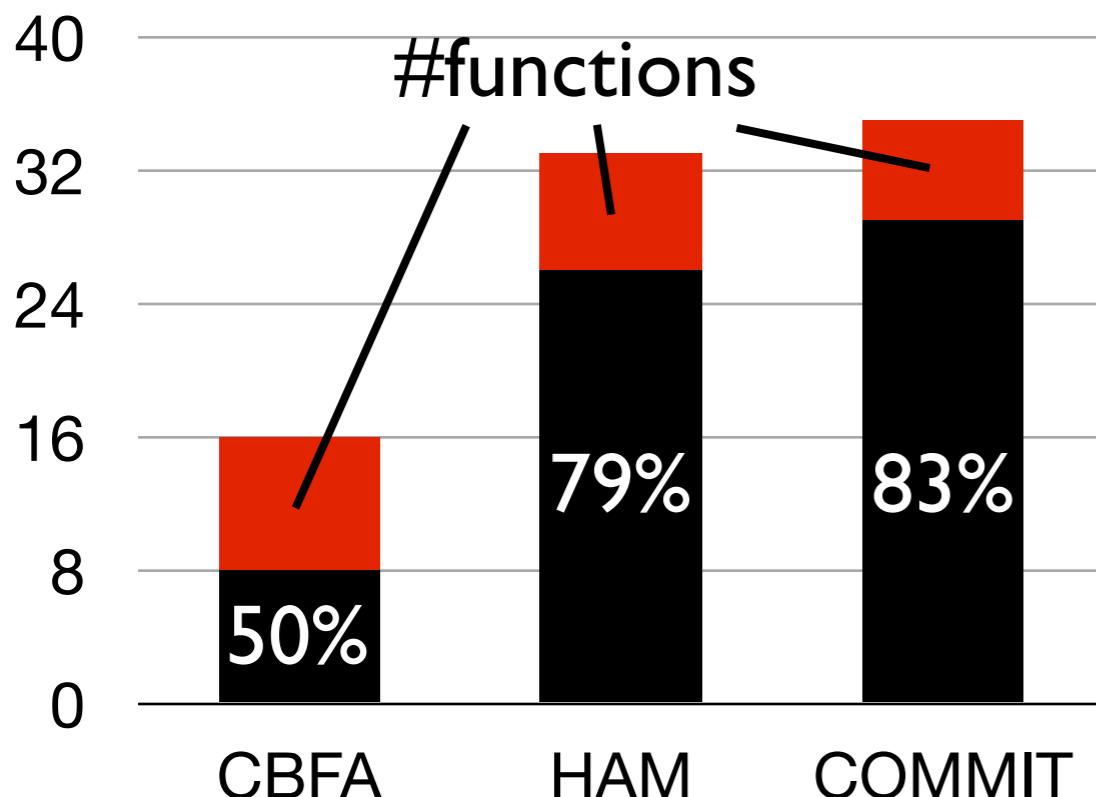
# H I. Richer Data Sources Yield richer Seeds



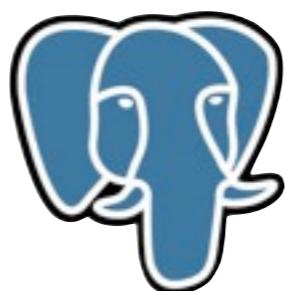
PostgreSQL



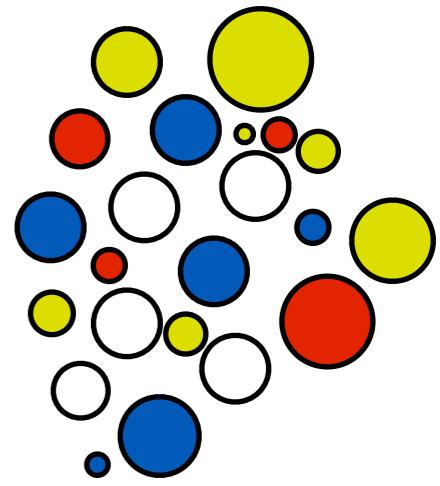
# H I. Richer Data Sources Yield richer Seeds



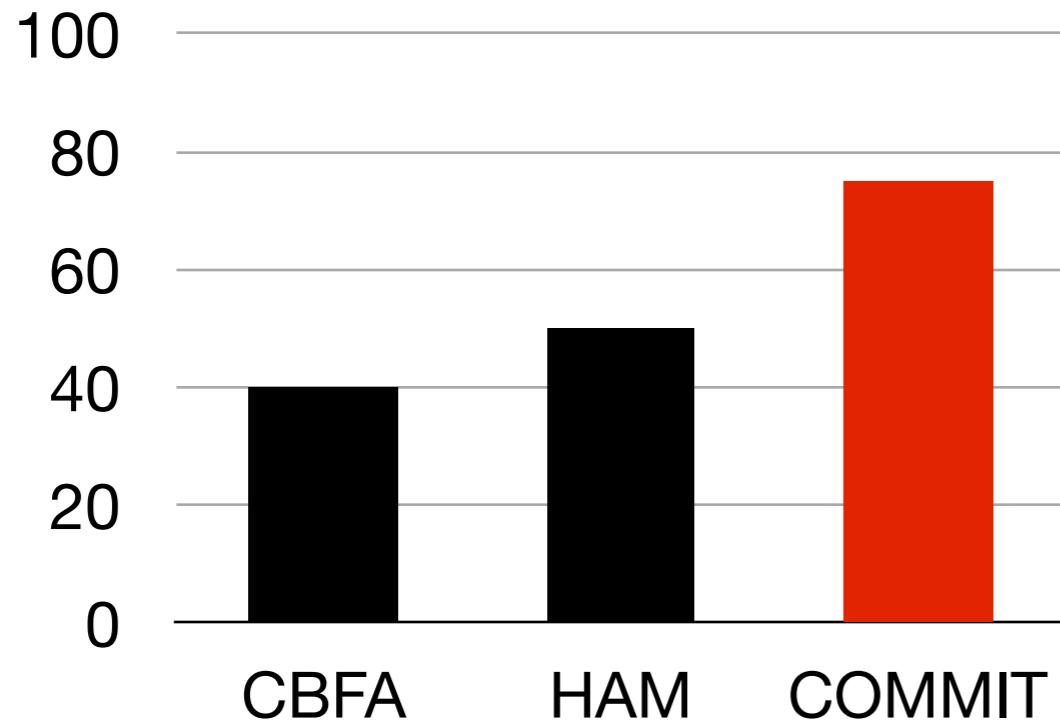
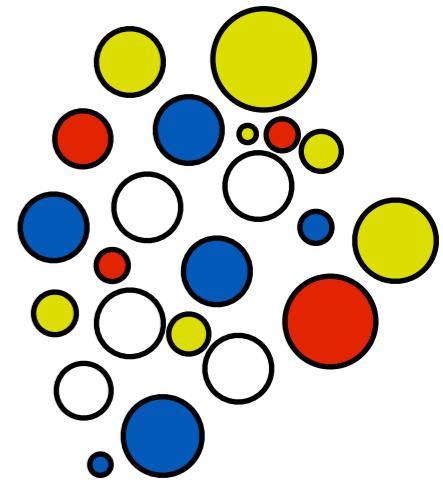
PostgreSQL



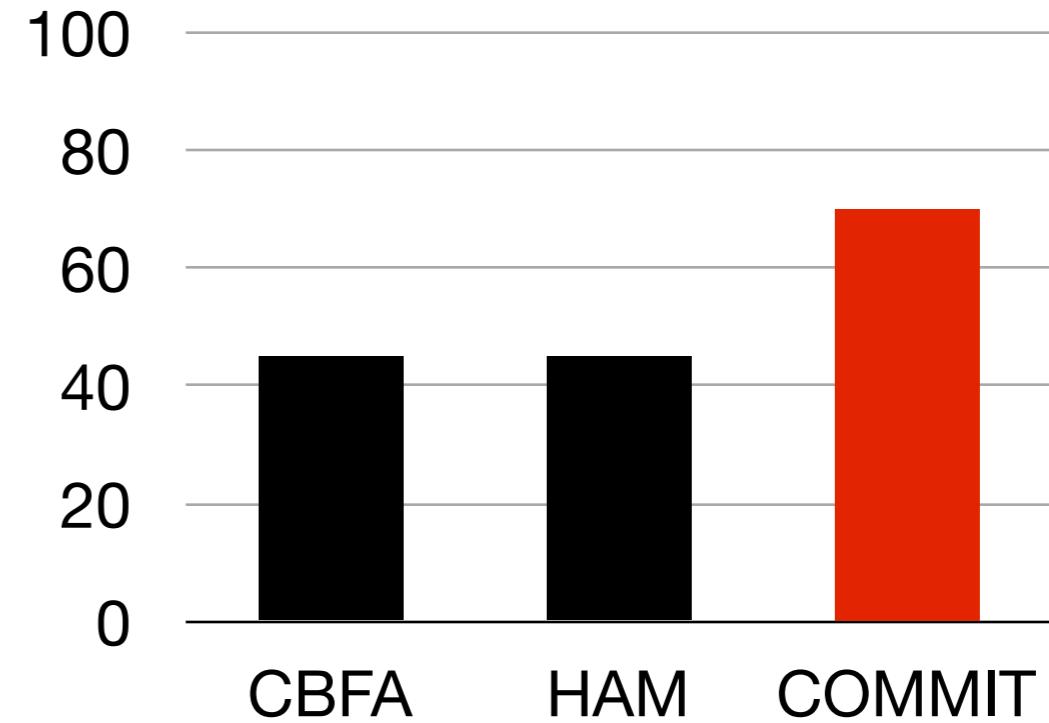
## H2. COMMIT Identifies a Larger Percentage of **unique** Concerns



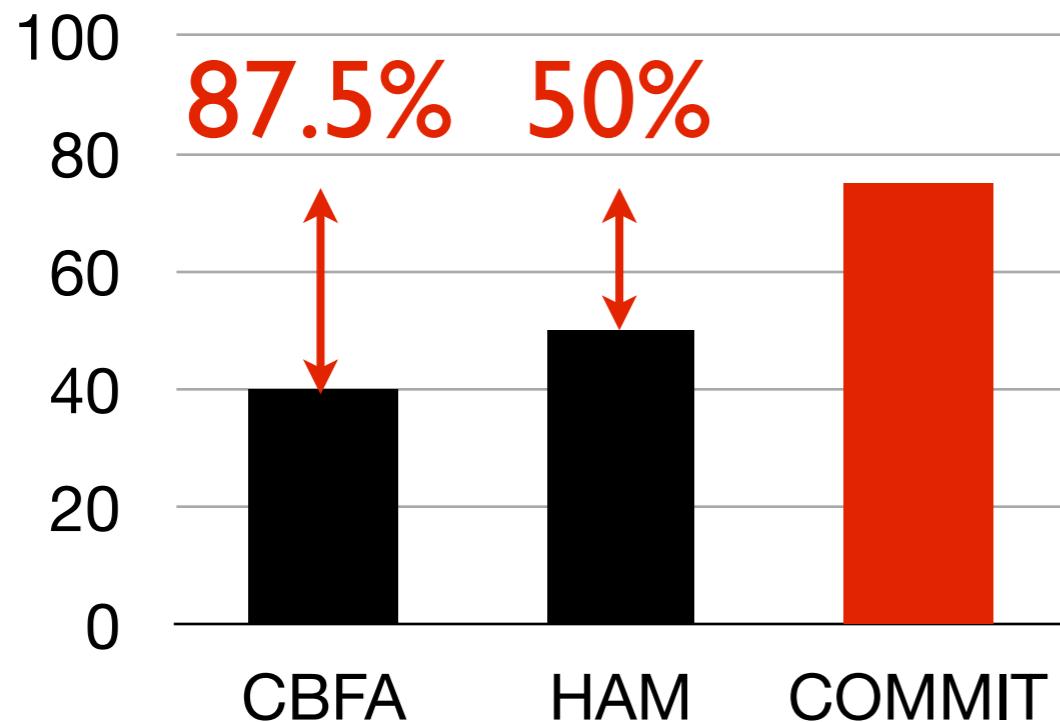
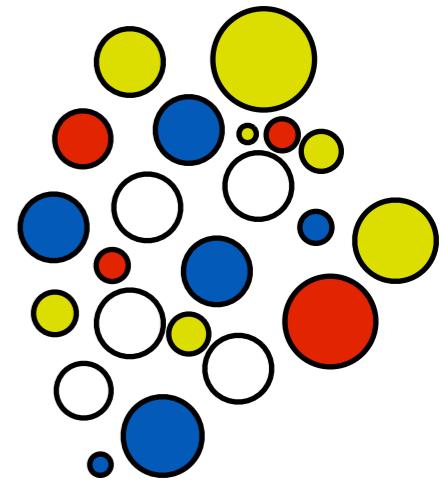
## H2. COMMIT Identifies a Larger Percentage of unique Concerns



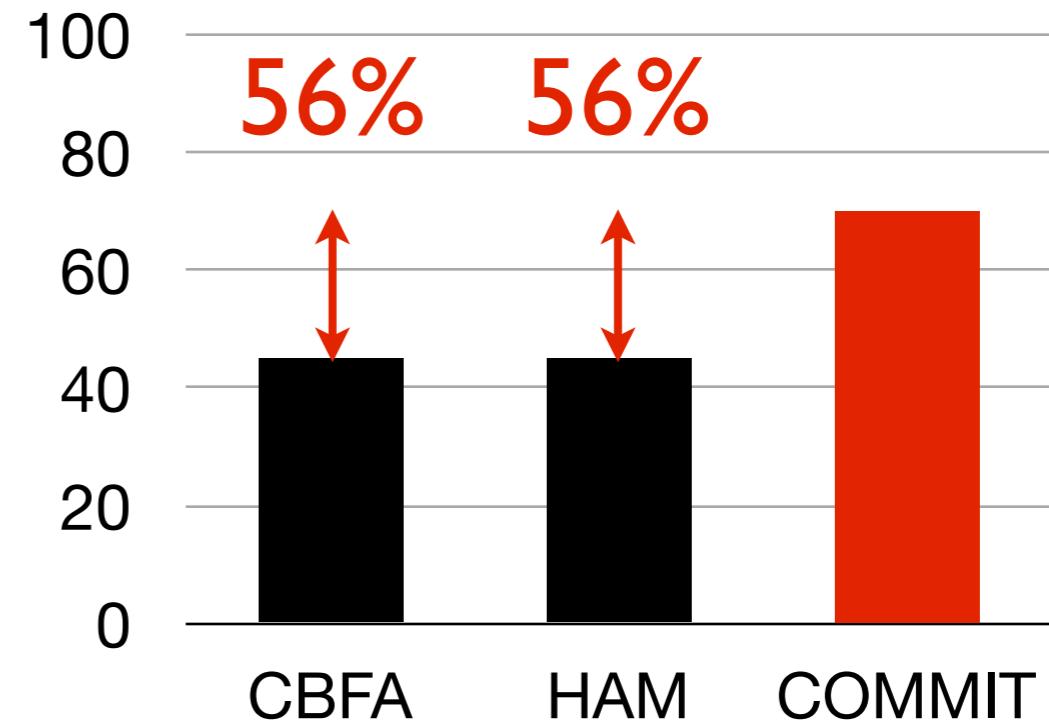
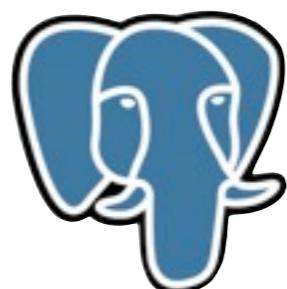
PostgreSQL



## H2. COMMIT Identifies a Larger Percentage of unique Concerns

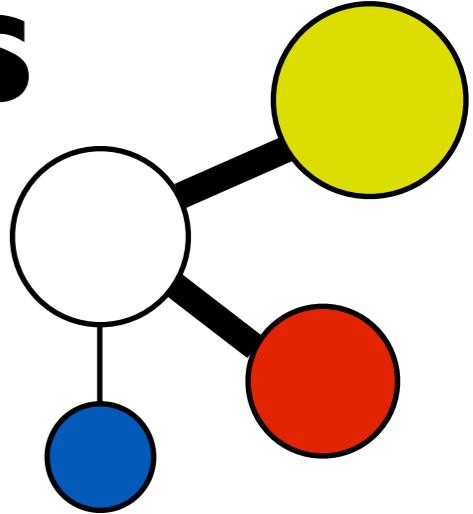


PostgreSQL



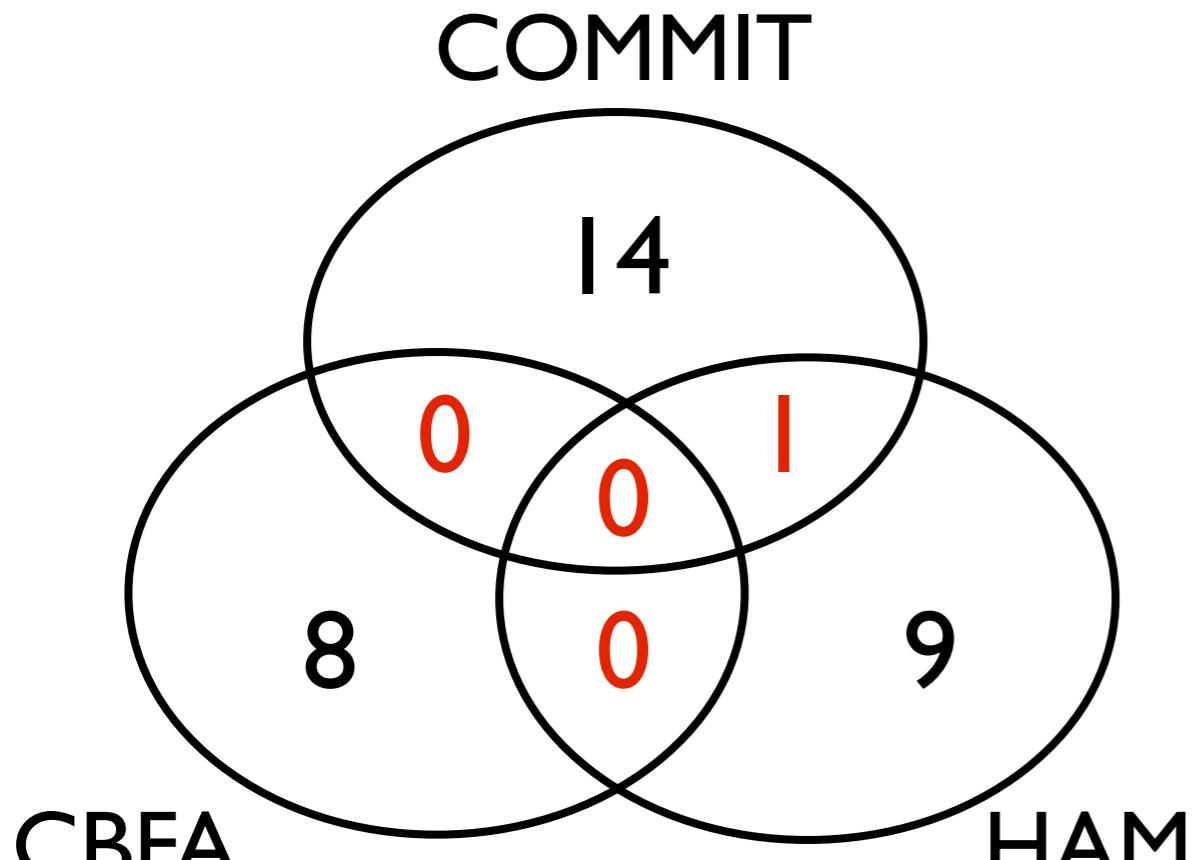
# H3. COMMIT complements

## CBFA and HAM (I)

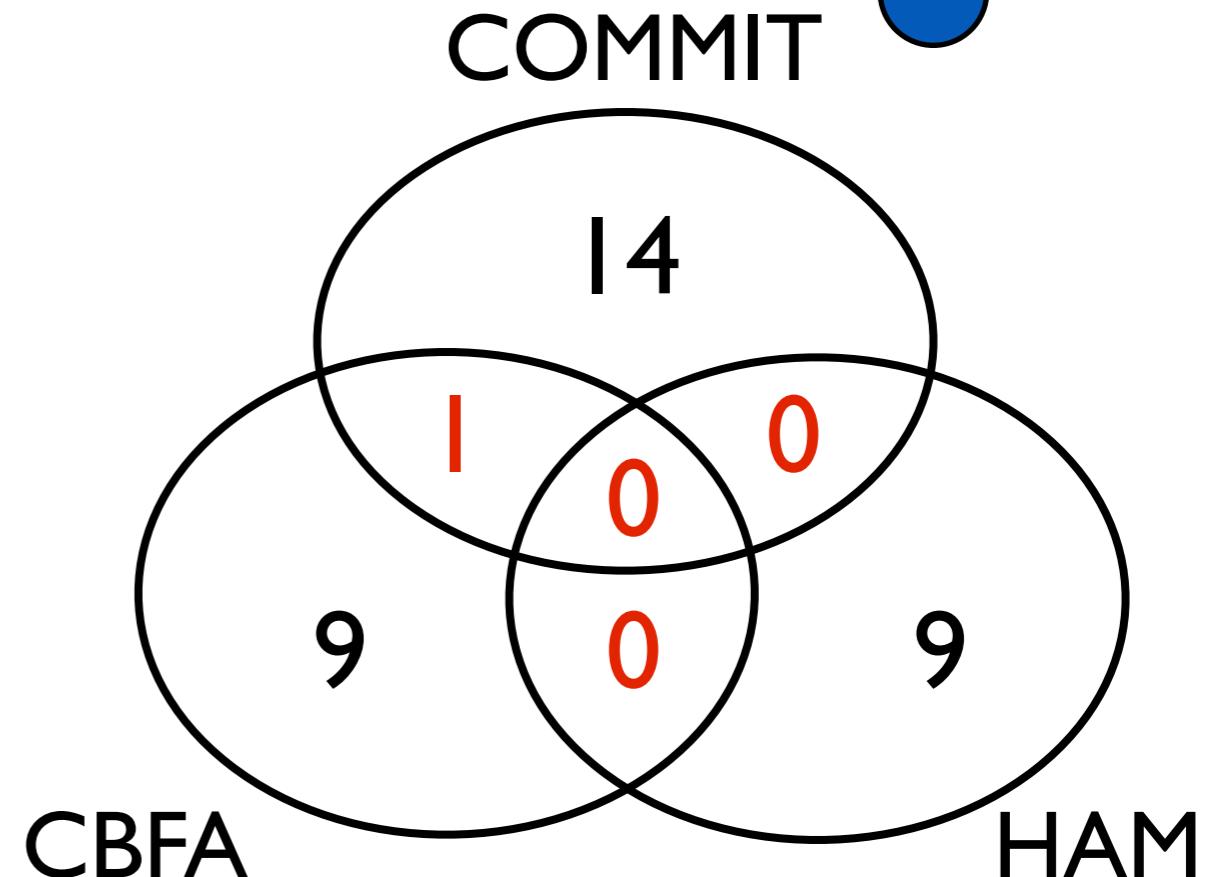


# H3. COMMIT complements

## CBFA and HAM (I)

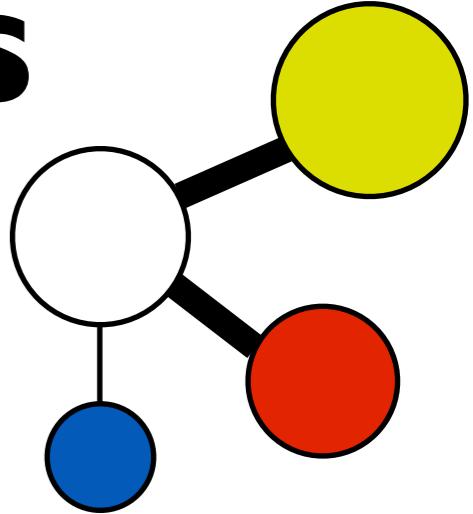


PostgreSQL



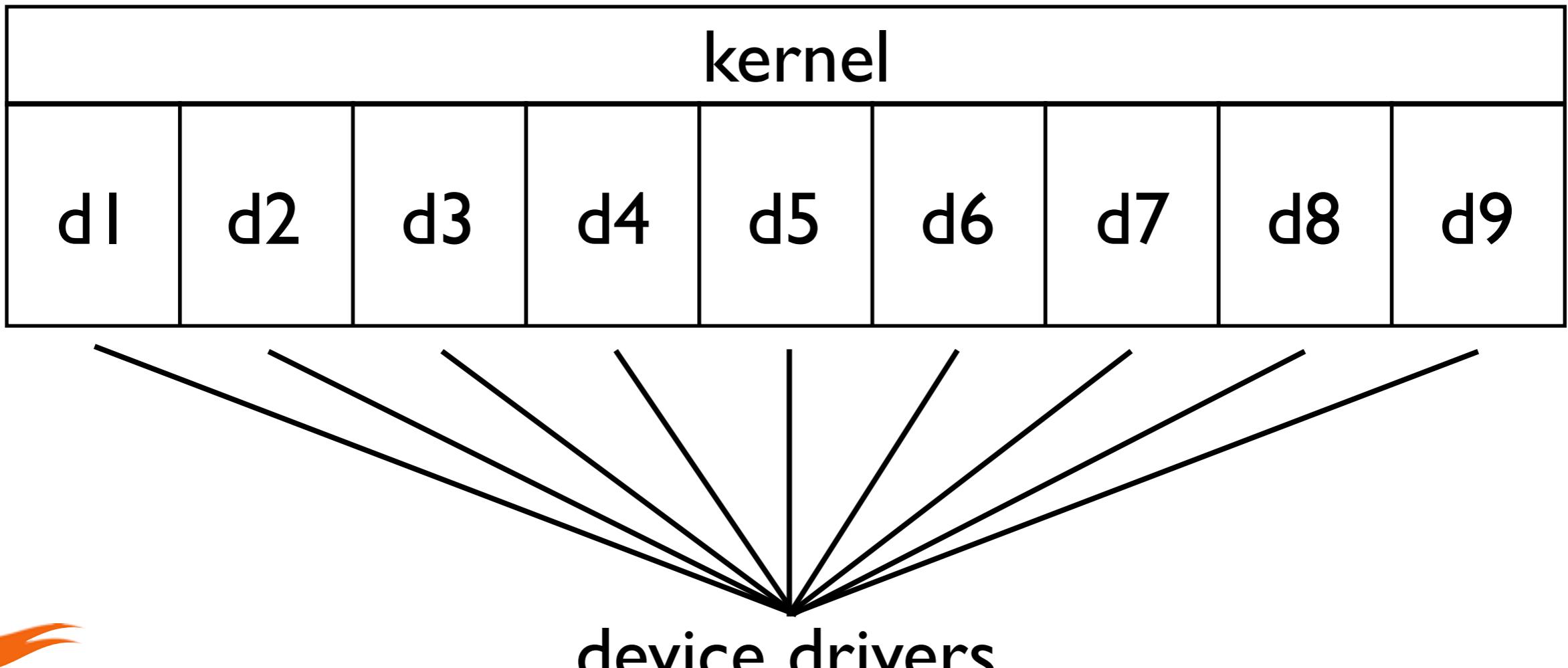
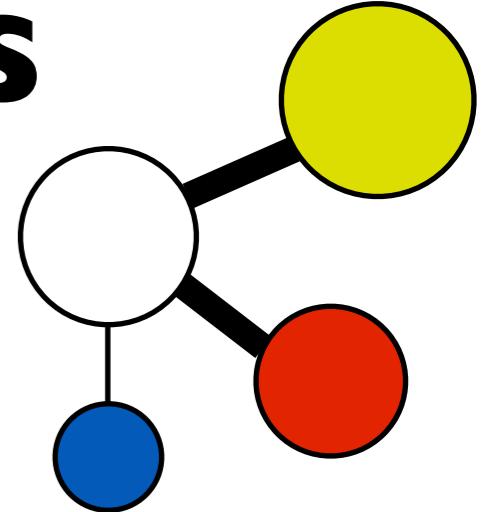
# H3. COMMIT complements

## CBFA and HAM (2)



# H3. COMMIT complements

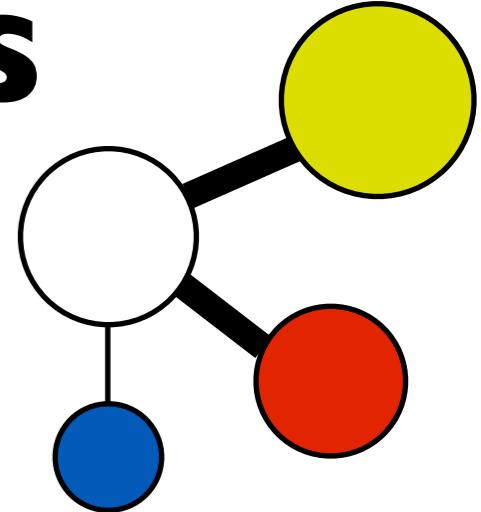
## CBFA and HAM (2)



# H3. COMMIT complements

## CBFA and HAM (2)

CBFA concern  
(e.g., driver API)

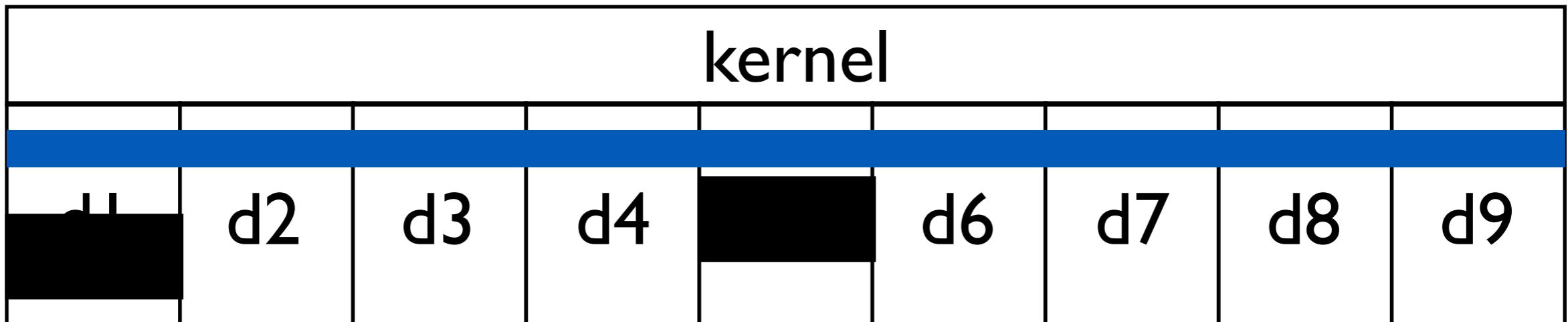
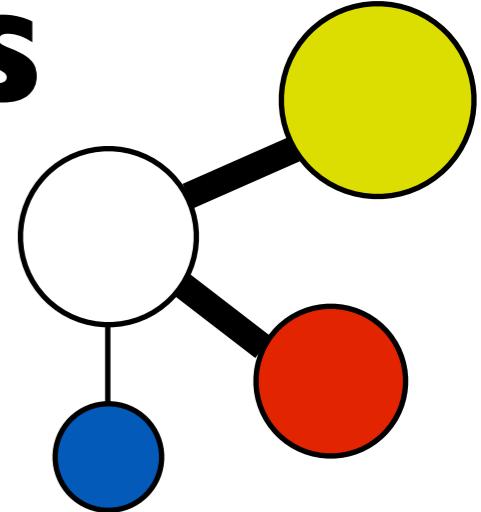


kernel								
d1	d2	d3	d4	d5	d6	d7	d8	d9

# H3. COMMIT complements

## CBFA and HAM (2)

CBFA concern  
(e.g., driver API)

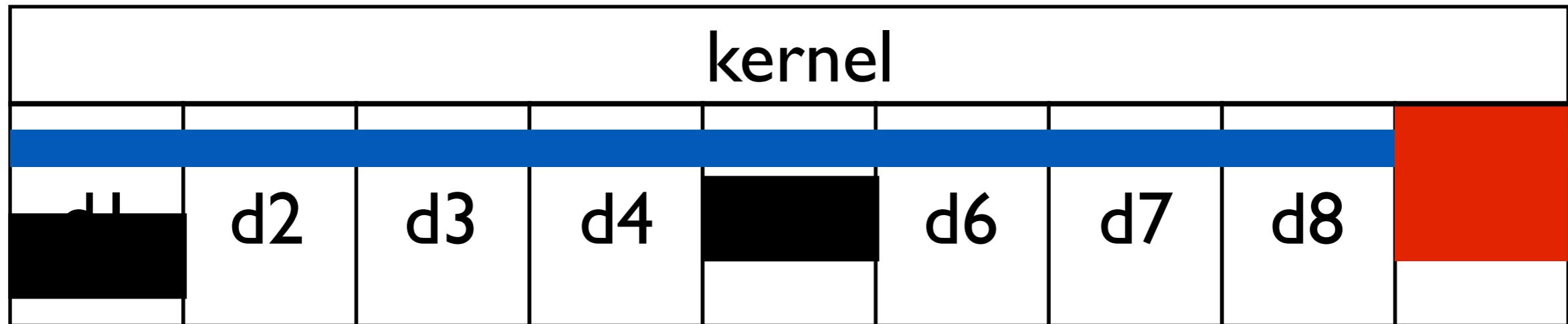
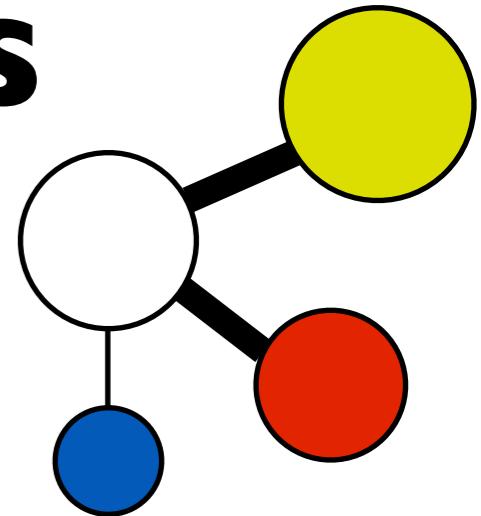


HAM concern  
(e.g., cloned driver code)

# H3. COMMIT complements

## CBFA and HAM (2)

CBFA concern  
(e.g., driver API)

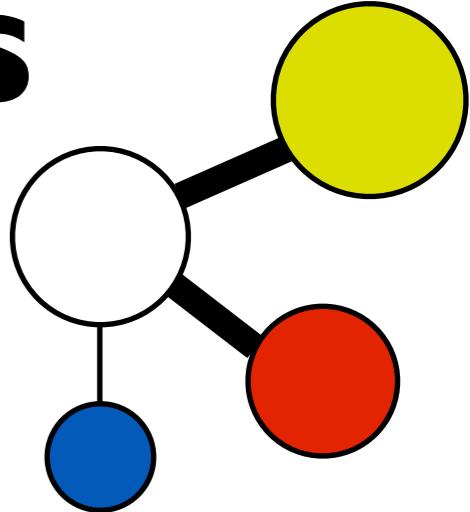


HAM concern  
(e.g., cloned driver code)

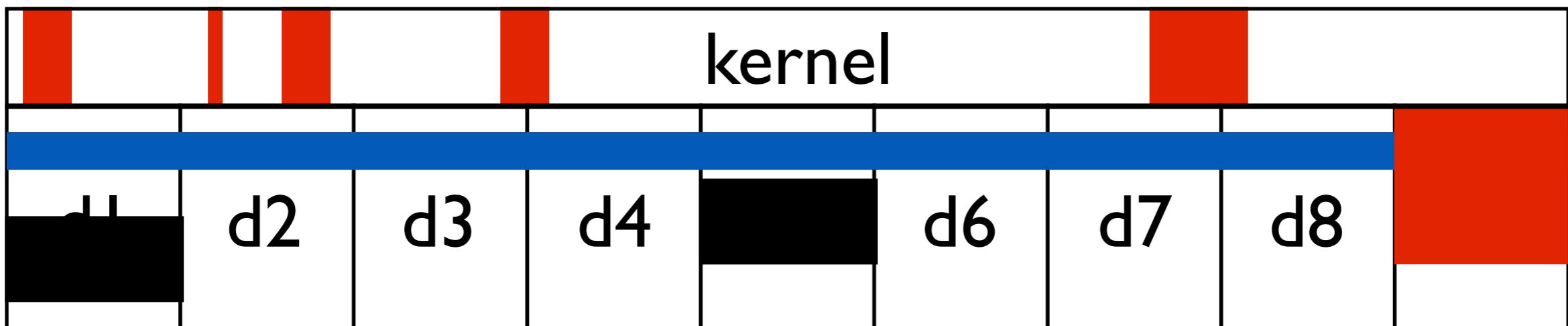
COMMIT concern  
(e.g., driver +  
infrastructure)

# H3. COMMIT complements

## CBFA and HAM (2)



CBFA concern  
(e.g., driver API)



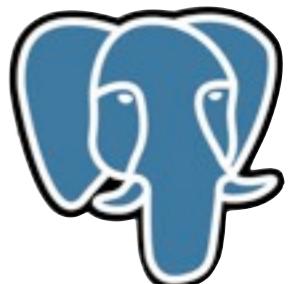
HAM concern  
(e.g., cloned driver code)

COMMIT concern  
(e.g., driver +  
infrastructure)

# **ODBC Data Retrieval**

## **Composite Concern**

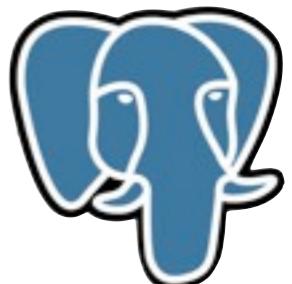
PostgreSQL



# ODBC Data Retrieval Composite Concern

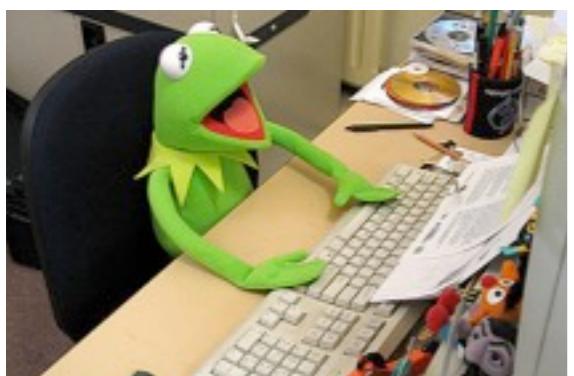
## I. connection configuration

PostgreSQL



# ODBC Data Retrieval Composite Concern

## I. connection configuration



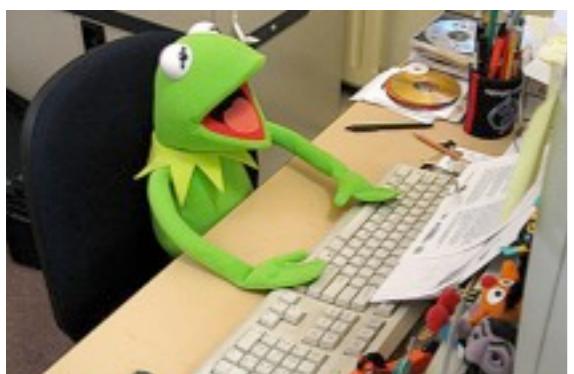
**ODBC**

Postgre<sup>SQL</sup>



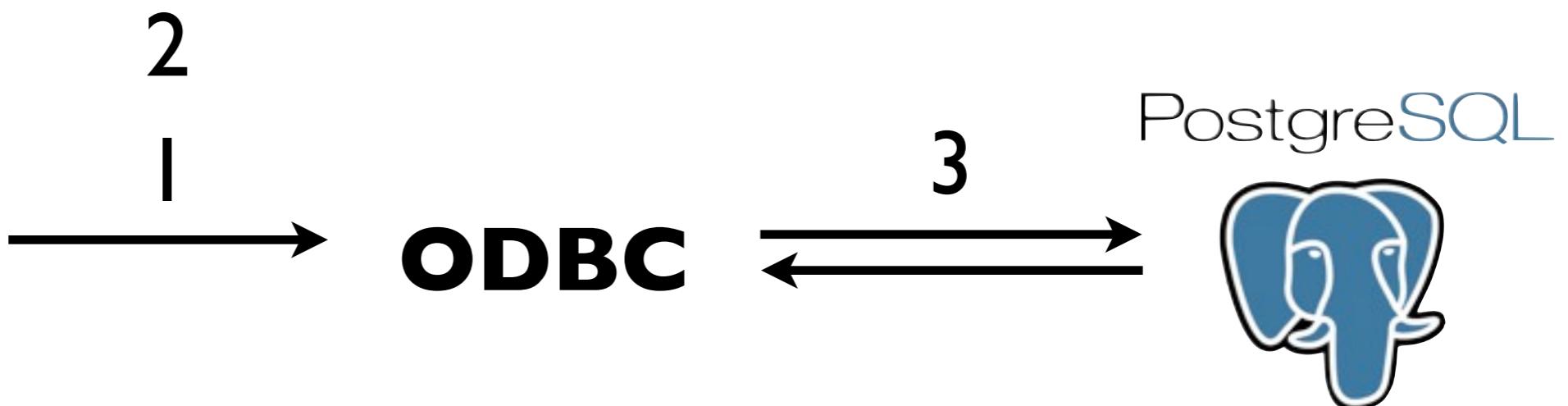
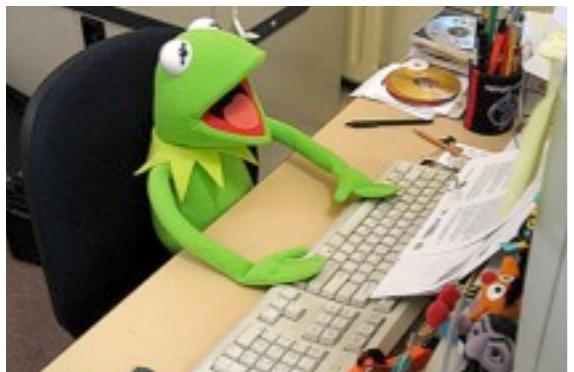
# ODBC Data Retrieval Composite Concern

1. connection configuration
2. connection error handling



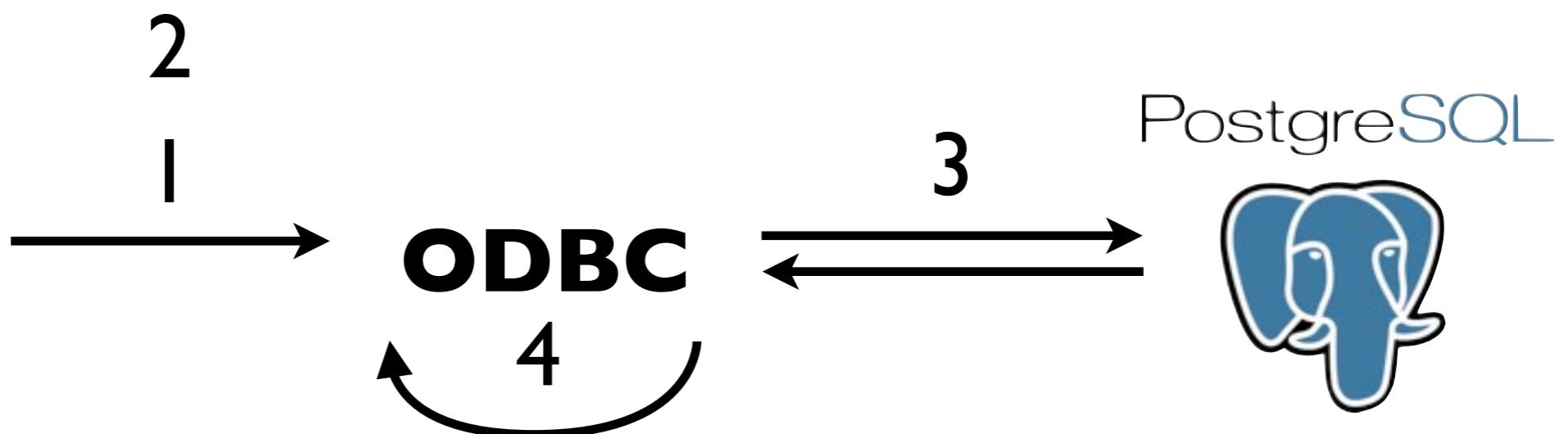
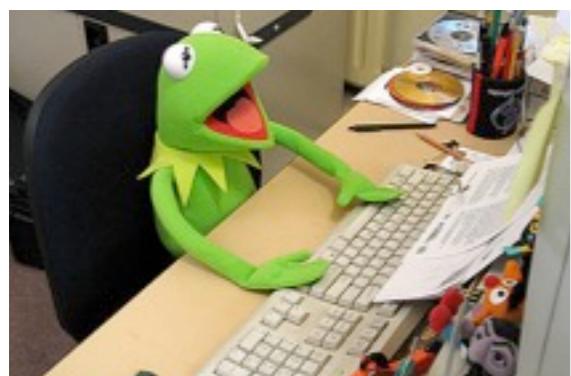
# ODBC Data Retrieval Composite Concern

1. connection configuration
2. connection error handling
3. data transfer



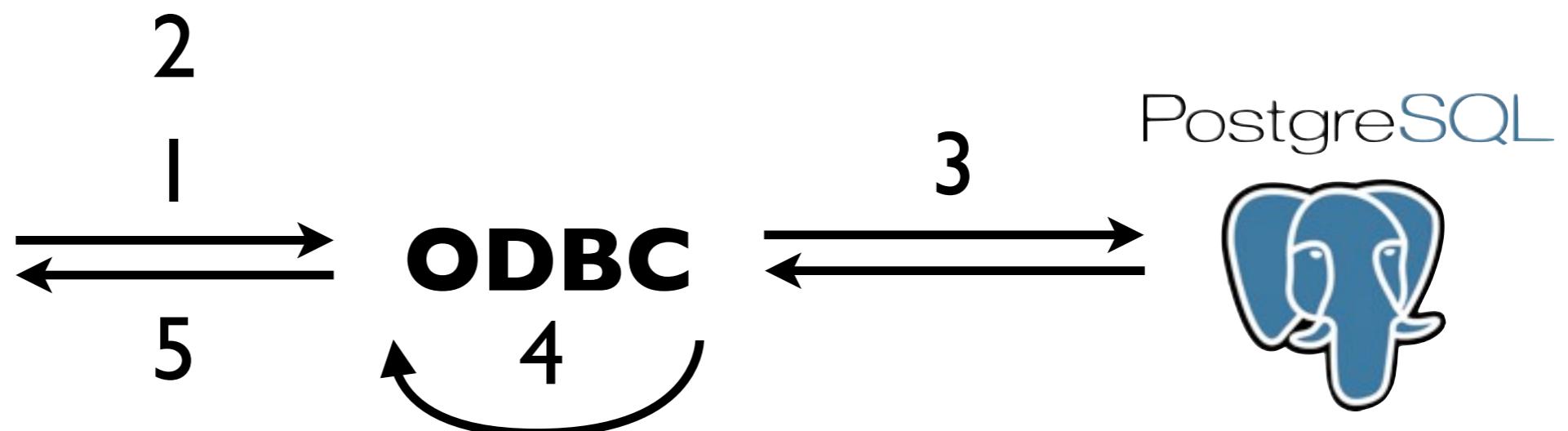
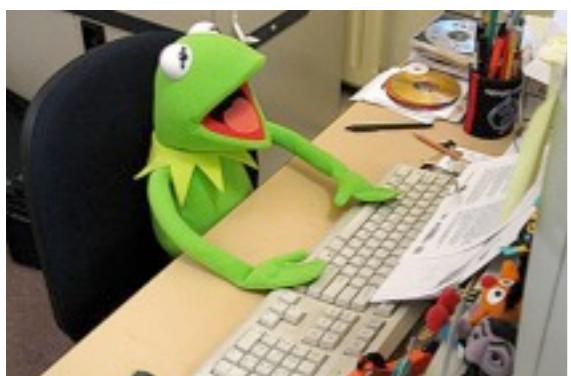
# ODBC Data Retrieval Composite Concern

1. connection configuration
2. connection error handling
3. data transfer
4. SQL-to-ODBC conversion



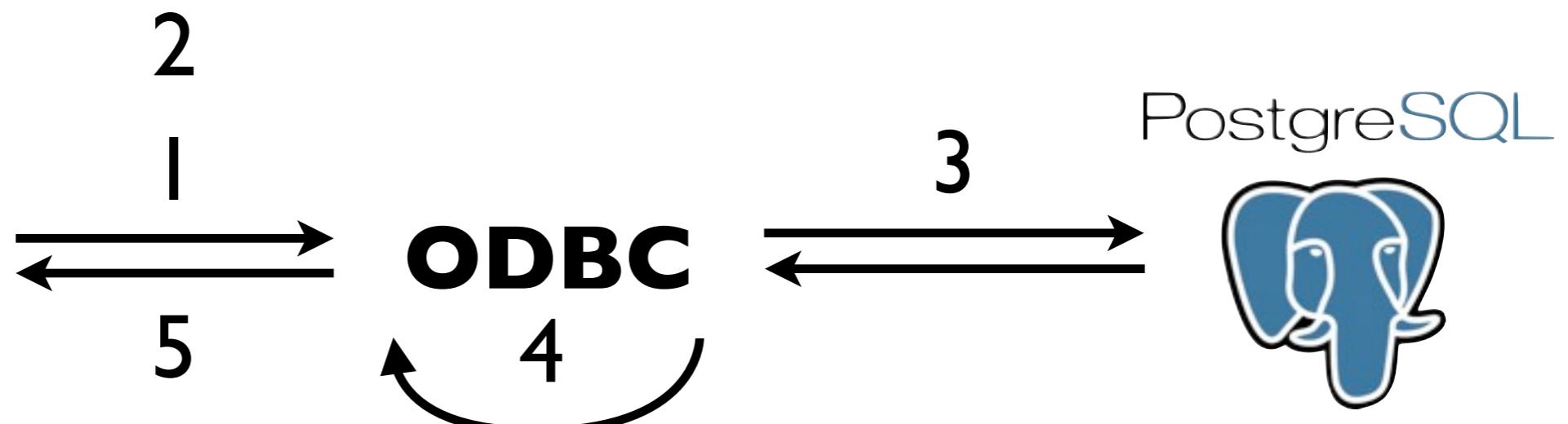
# ODBC Data Retrieval Composite Concern

1. connection configuration
2. connection error handling
3. data transfer
4. SQL-to-ODBC conversion
5. ODBC-to-ESQL conversion



# ODBC Data Retrieval Composite Concern

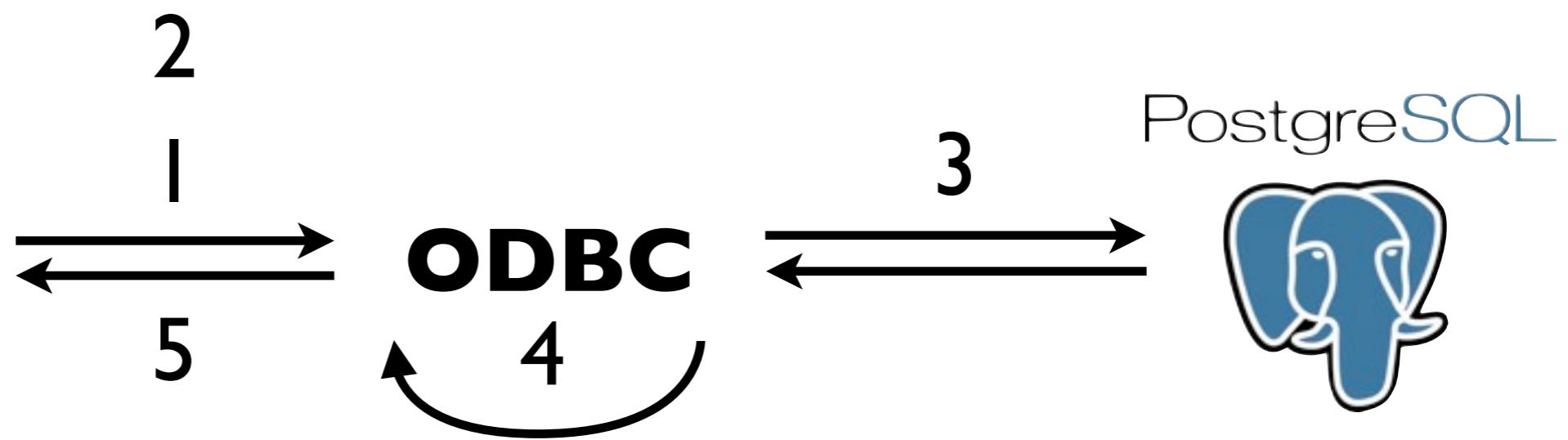
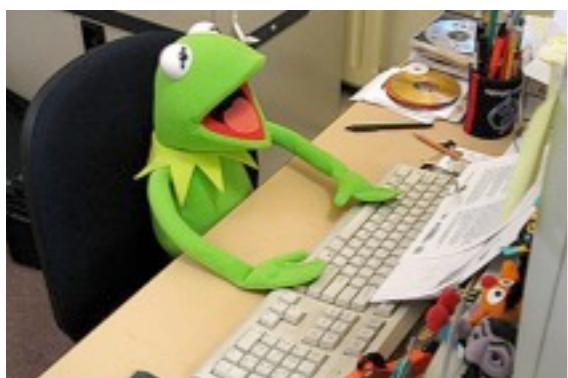
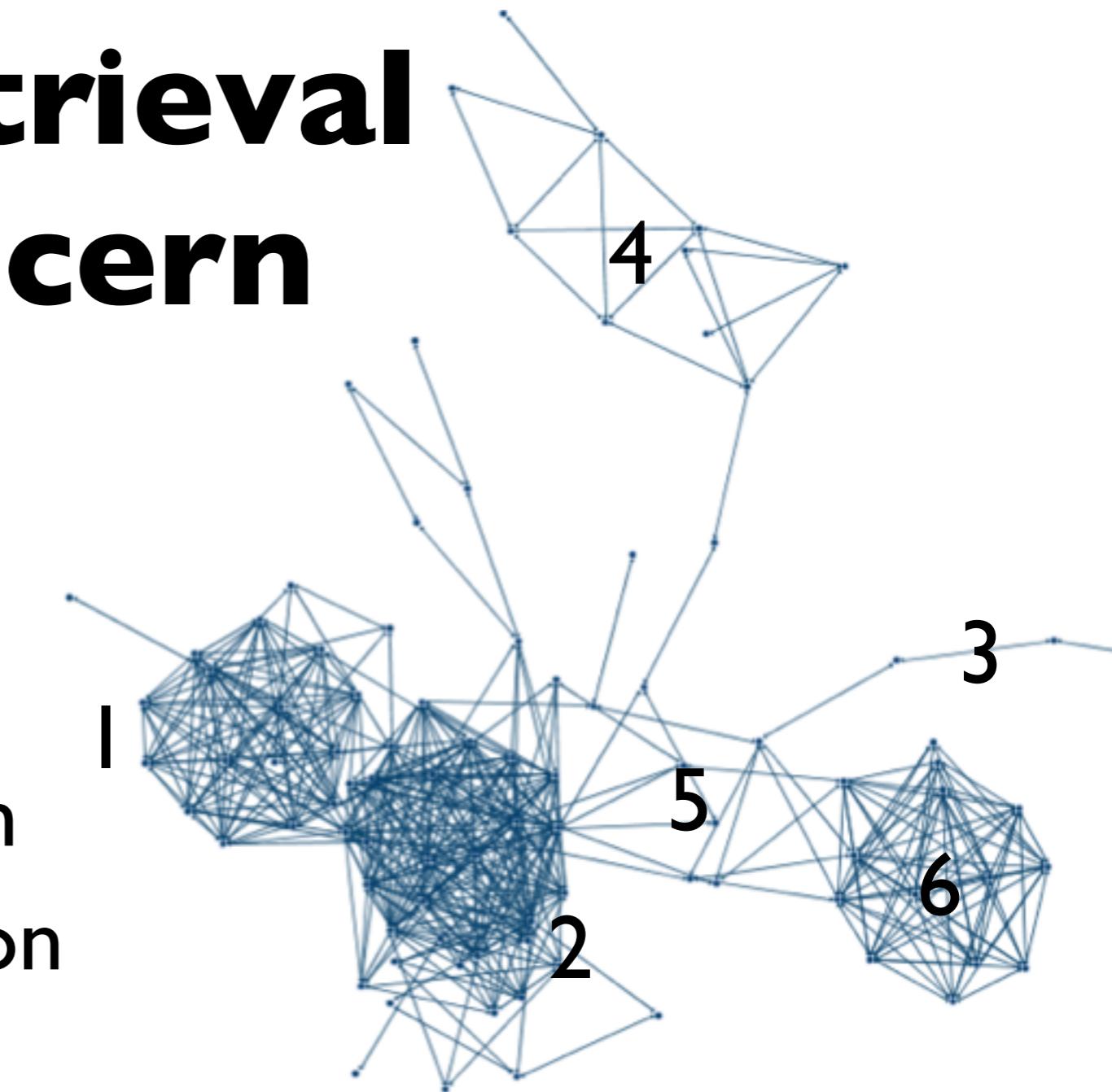
1. connection configuration
2. connection error handling
3. data transfer
4. SQL-to-ODBC conversion
5. ODBC-to-ESQL conversion
6. conversion error handling



# ODBC Data Retrieval

## Composite Concern

1. connection configuration
2. connection error handling
3. data transfer
4. SQL-to-ODBC conversion
5. ODBC-to-ESQL conversion
6. conversion error handling

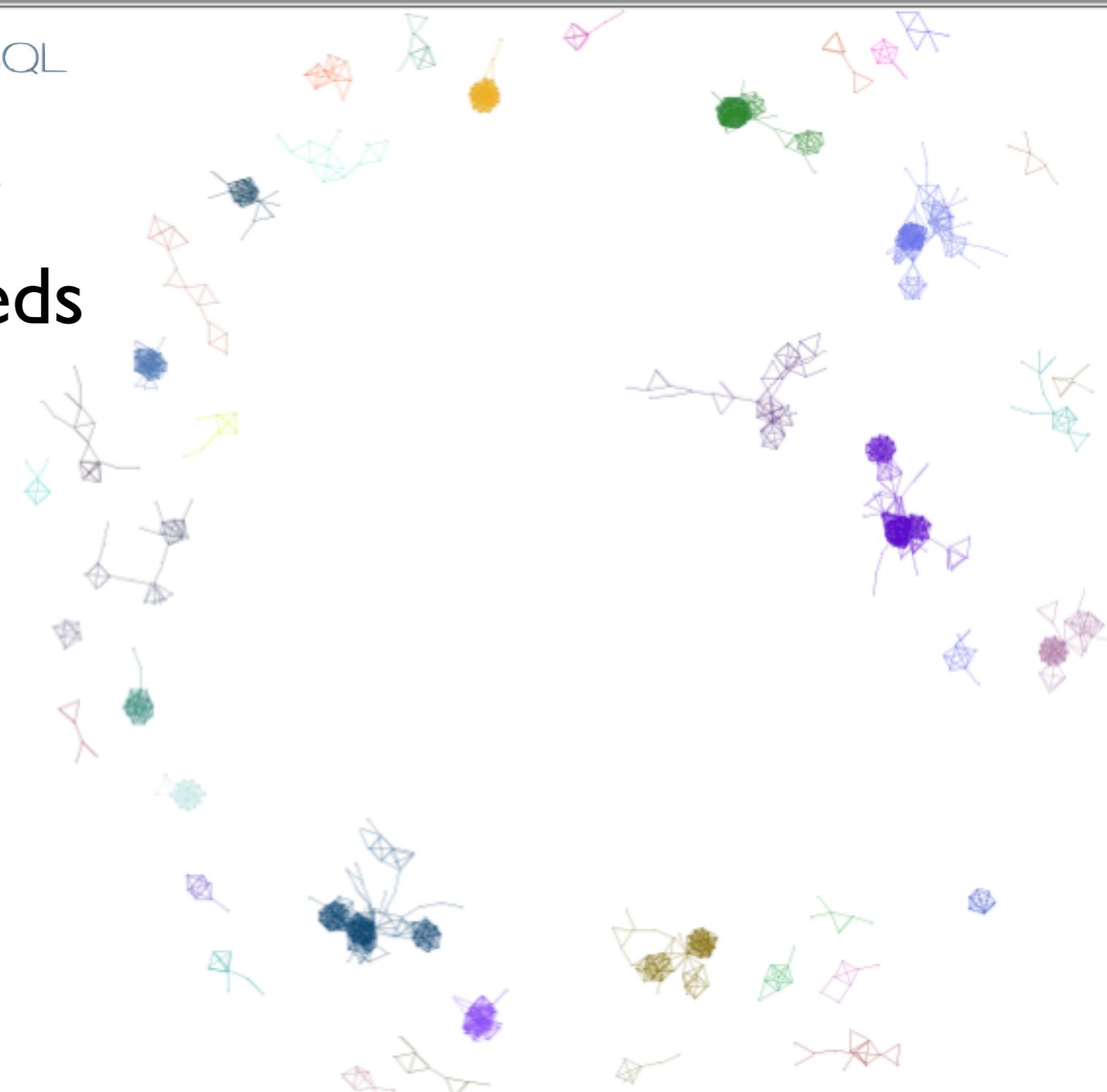


PostgreSQL



36 seeds

- grant\_statement\_processing
- access\_control
- low\_level\_memory\_allocation
- time\_decoding
- io\_with\_authentication
- table\_insertion
- table\_locking
- postgresql\_transaction\_command
- sql\_constraint\_checking
- bit\_manipulation
- buffer\_file\_io
- order\_by
- buffer\_hashmap
- odbc\_query\_exception\_handling
- odbc\_type\_conversion
- tuple\_sorting
- transaction\_log\_data\_management
- grant\_unifying\_configuration
- transaction\_log\_navigation
- spi\_management
- time\_formatting
- pl\_python\_exception\_handling
- gui\_global\_parameters
- lock\_debugging
- query\_planner\_config
- sql\_query\_exception\_handling
- query\_invocation
- encoding\_conversion
- index\_update
- shared\_memory\_management
- execution\_plan\_manipulation
- alignment
- ipc\_synchronisation
- replay\_transaction\_logs
- odbc\_property\_config
- plpgsql\_debugging
- database\_statistics
- execution\_plan
- ident\_authentication\_management
- embedded\_language\_scanners
- client\_server\_messages



call

function

file

subsystem

concern

CENTER

Interpreter

Control Panel

Select a state





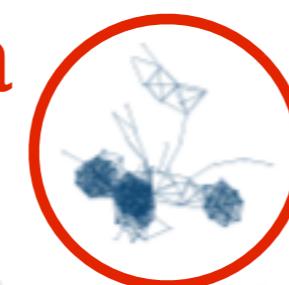
PostgreSQL



36 seeds

- grant\_statement\_processing
- access\_control
- low\_level\_memory\_allocation
- time\_decoding
- io\_with\_authentication
- table\_insertion
- table\_locking
- postgresql\_transaction\_command
- sql\_constraint\_checking
- bit\_manipulation
- buffer\_file\_io
- order\_by
- buffer\_hashmap
- odbc\_query\_exception\_handling
- odbc\_type\_conversion
- tuple\_sorting
- transaction\_log\_data\_management
- grant\_unifying\_configuration
- transaction\_log\_navigation
- spi\_management
- time\_formatting
- pl\_python\_exception\_handling
- gui\_global\_parameters
- lock\_debugging
- query\_planner\_config
- sql\_query\_exception\_handling
- query\_invocation
- encoding\_conversion
- index\_update
- shared\_memory\_management
- execution\_plan\_manipulation
- alignment
- ipc\_synchronisation
- replay\_transaction\_logs
- odbc\_property\_config
- plpgsql\_debugging
- database\_statistics
- execution\_plan
- ident\_authentication\_management
- embedded\_language\_scanners
- client\_server\_messages

# ODBC Data Retrieval Concern



call

function

file

subsystem

concern

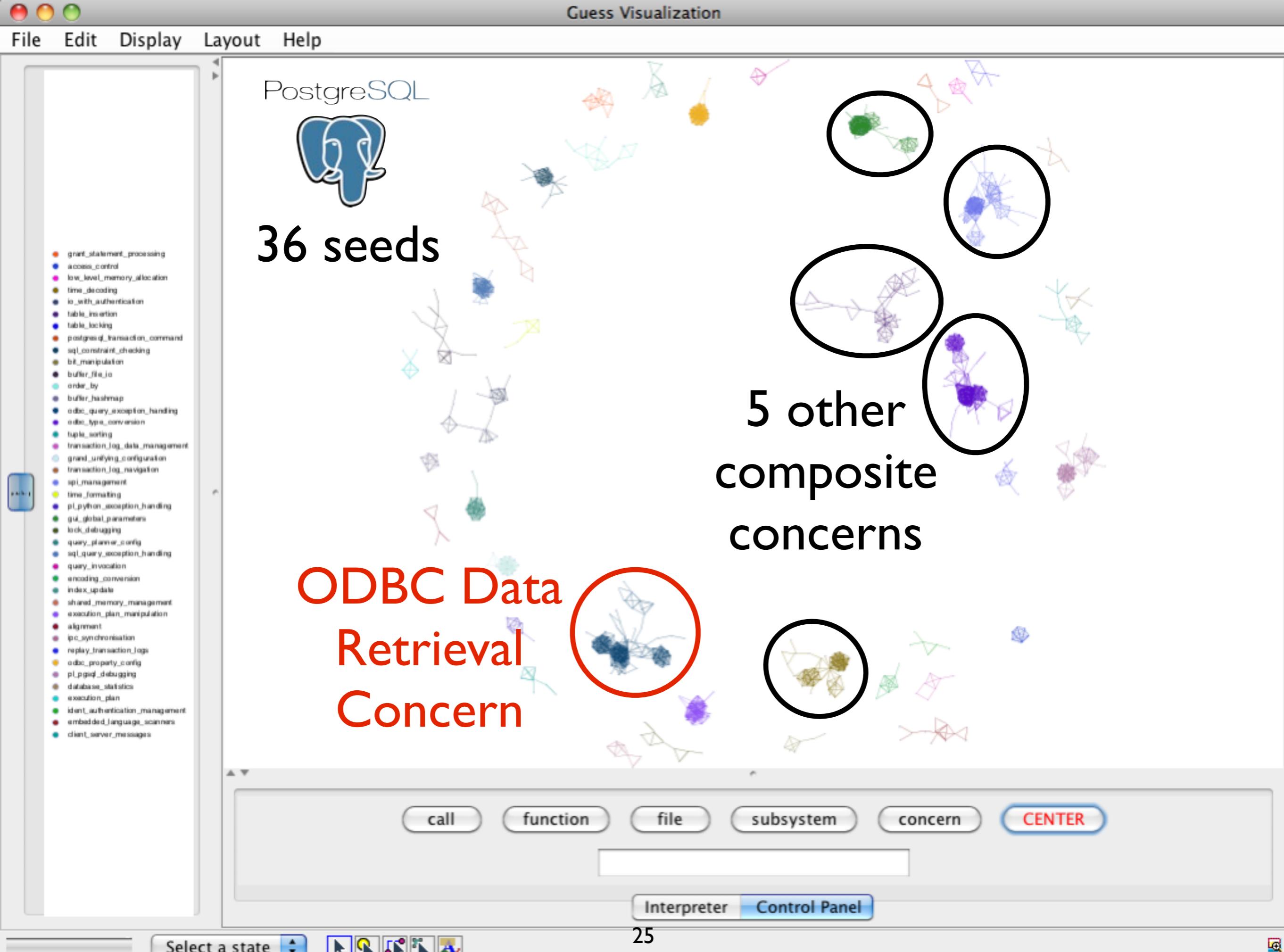
CENTER

Interpreter

Control Panel

Select a state





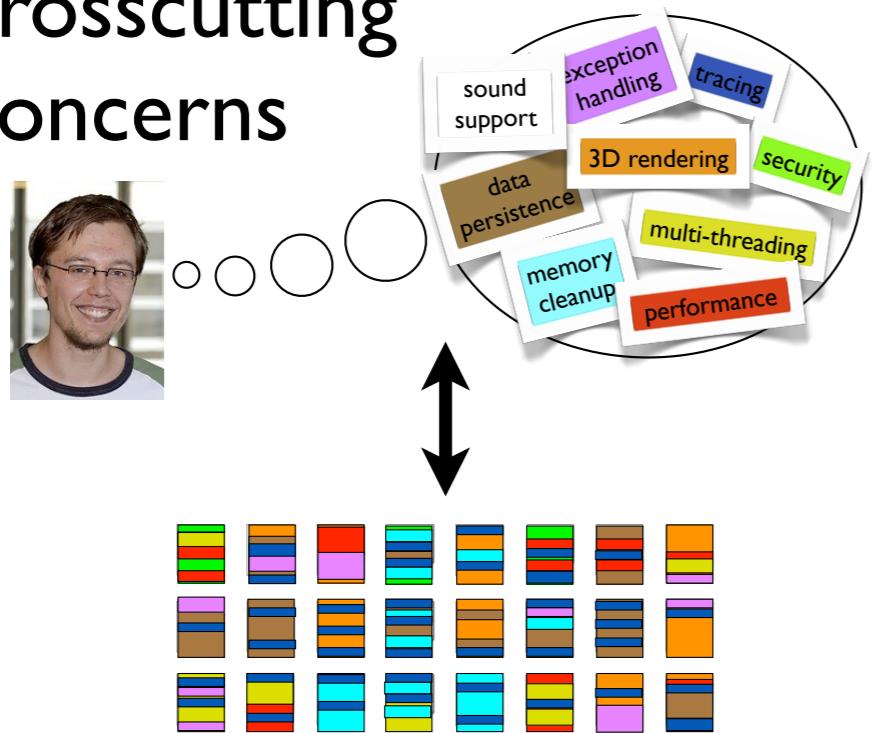
# Threats to Validity

- **generalizability** to other systems
- **subjectivity** ↔ substantial agreement (Kappa)
- seed quality not checked
- threshold optimization is **task-specific**

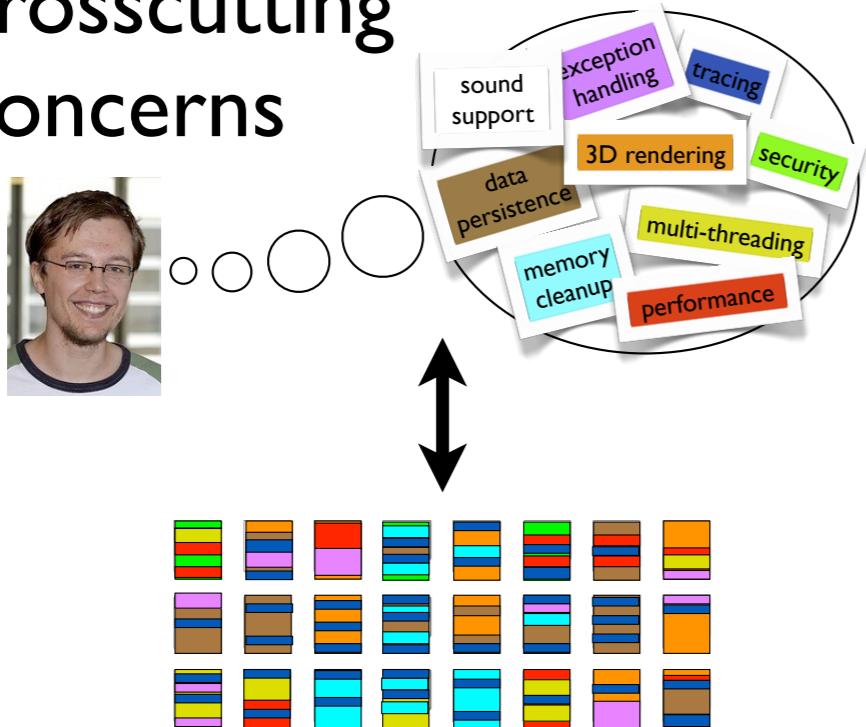
- 1. What is a Crosscutting Concern?**
- 2. The Concern Mining Process and its Shortcomings**
- 3. COMMIT**
- 4. Case Study**
- 5. Conclusion**



# Crosscutting Concerns



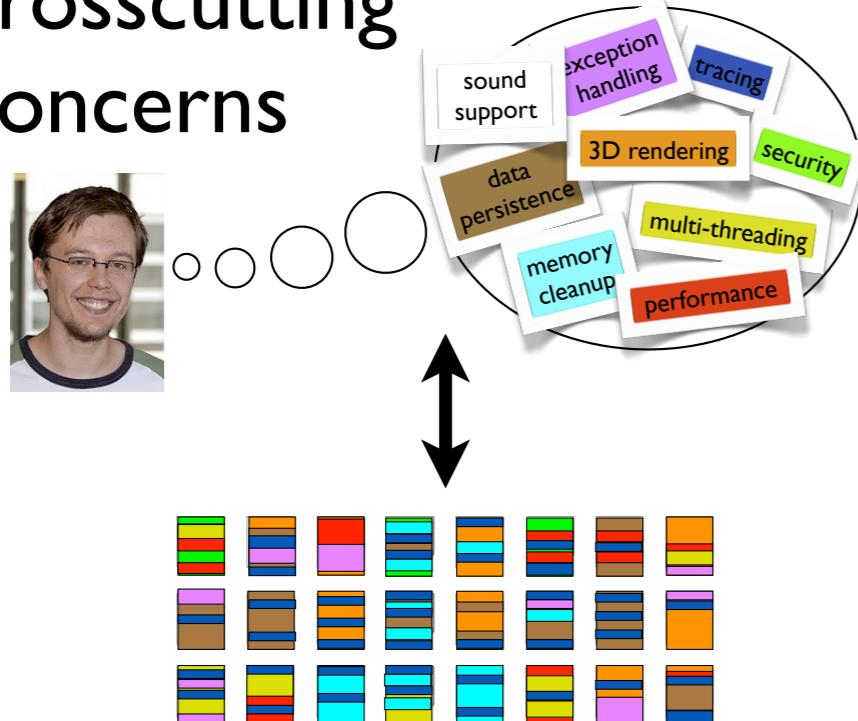
# Crosscutting Concerns



# Concern Mining Shortcomings

- S1. limited seed **context**
- S2. **noise** between seeds
- S3. no **composition** of concerns

# Crosscutting Concerns



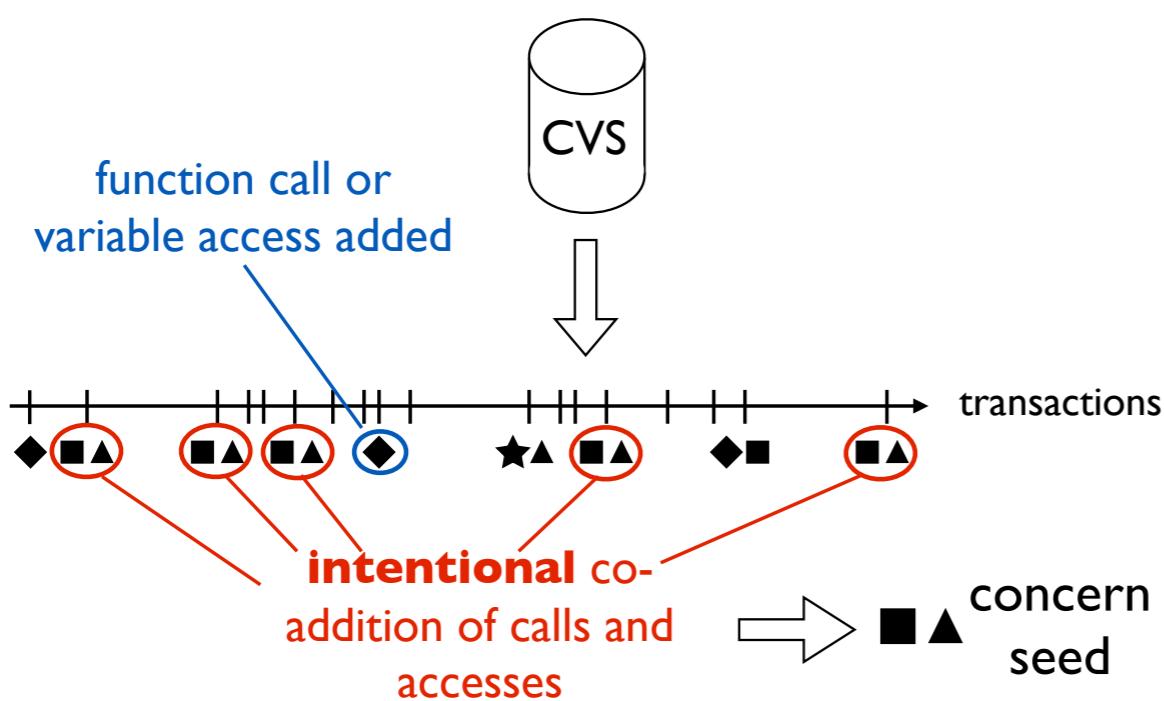
# Concern Mining Shortcomings

S1. limited seed **context**

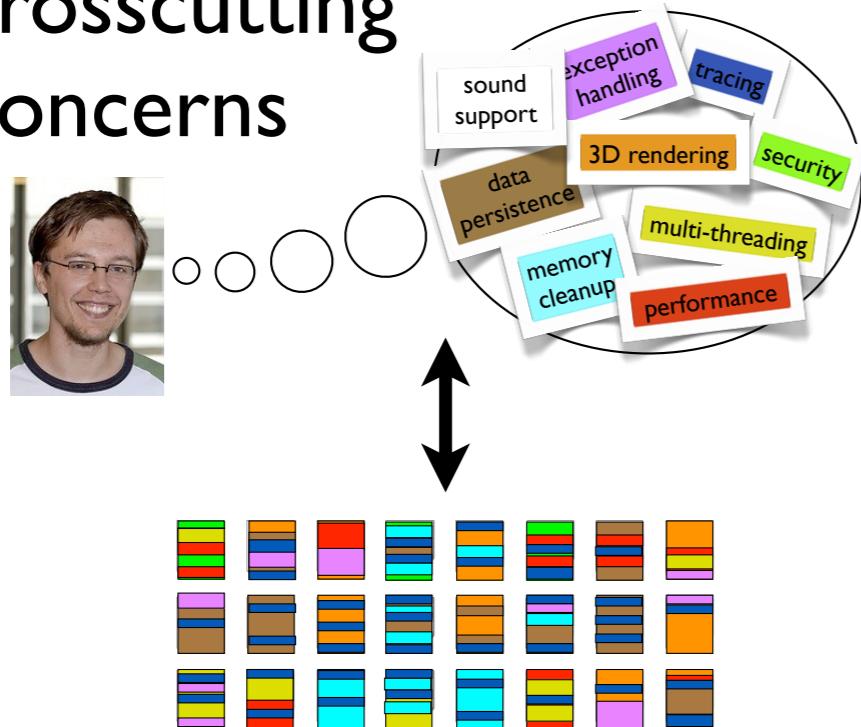
S2. **noise** between seeds

S3. no **composition** of concerns

## COMMIT

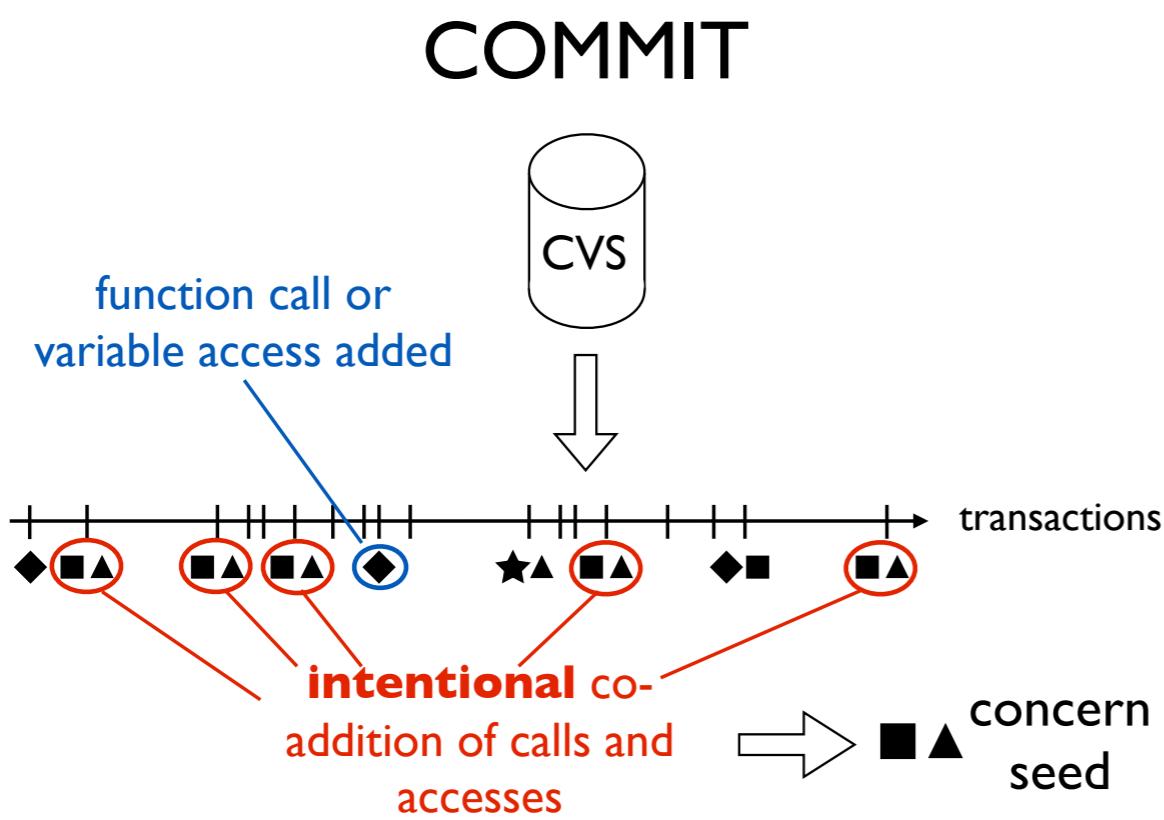


# Crosscutting Concerns

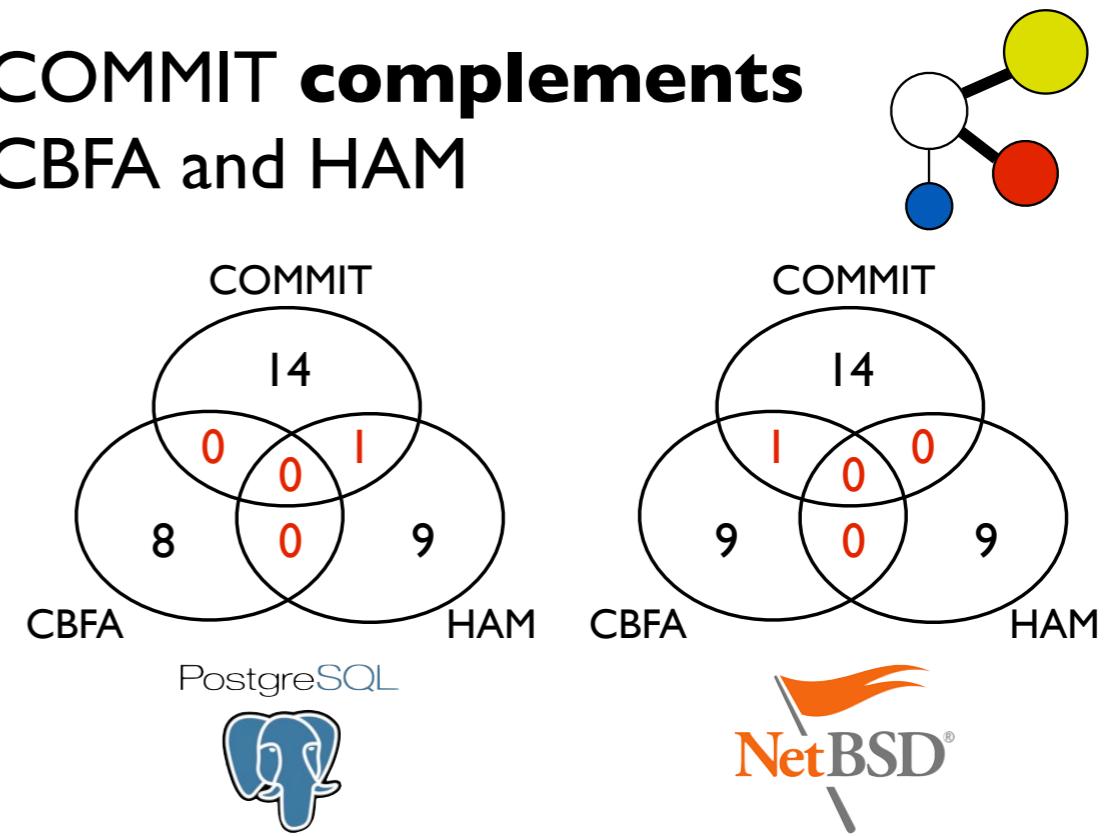


# Concern Mining Shortcomings

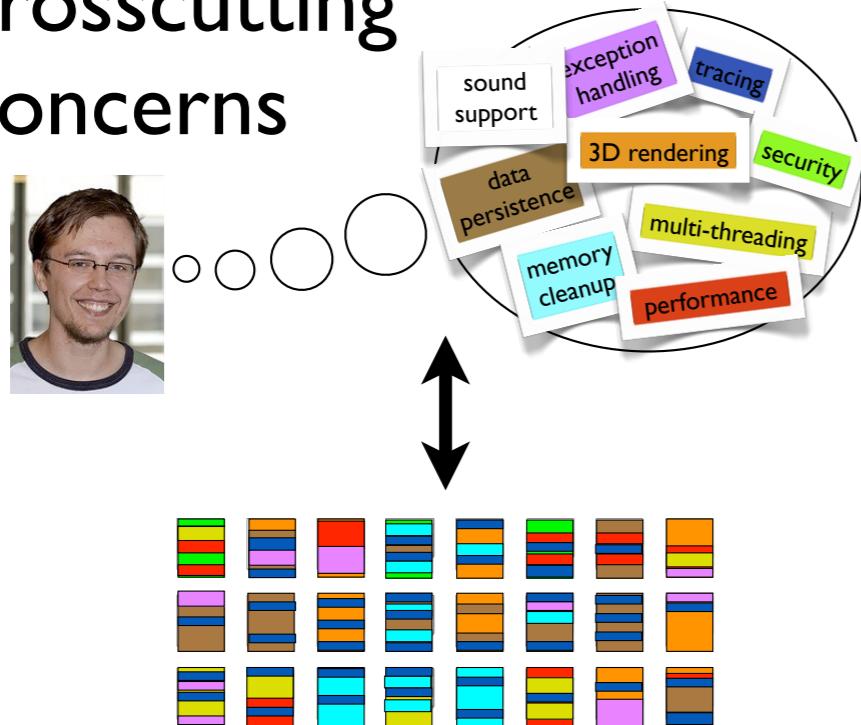
- S1. limited seed **context**
- S2. **noise** between seeds
- S3. no **composition** of concerns



## COMMIT complements CBFA and HAM



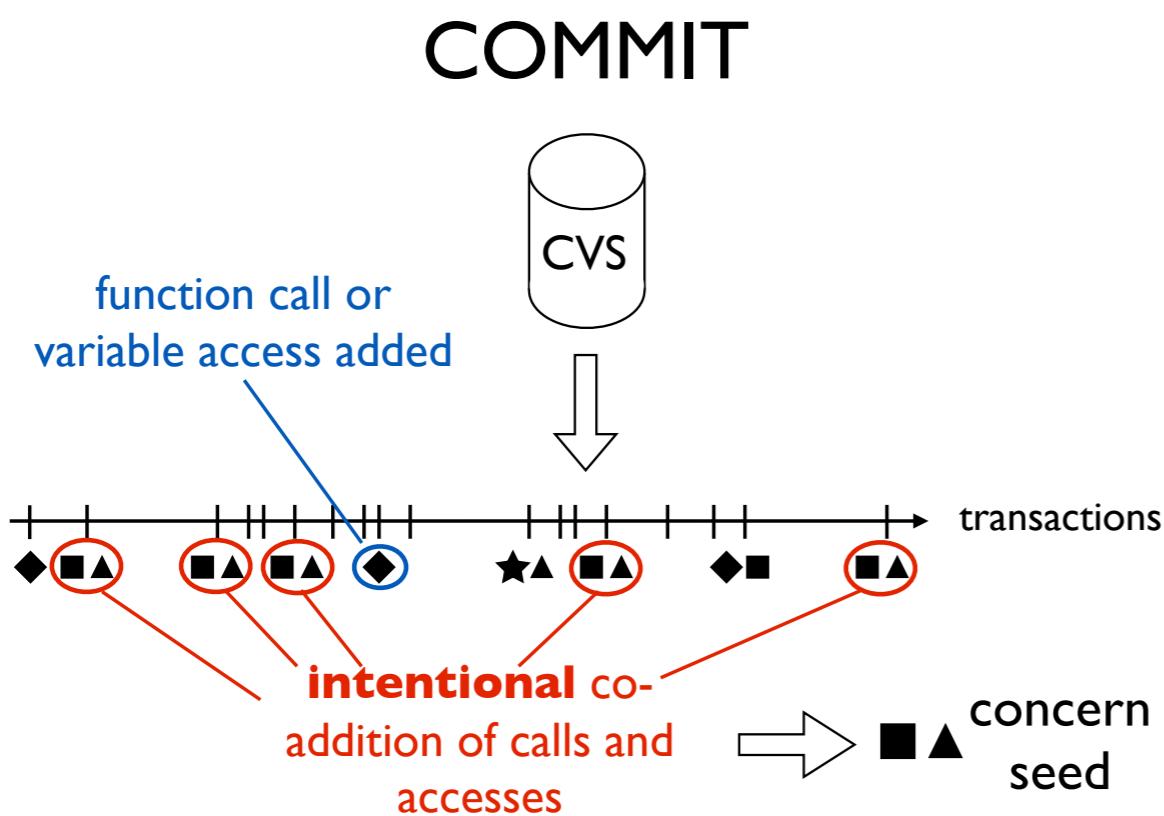
# Crosscutting Concerns



# Concern Mining Shortcomings

- S1. limited seed **context**
- S2. **noise** between seeds
- S3. no **composition** of concerns

# QUESTIONS?



## COMMIT complements CBFA and HAM

