



Accelerating Development/Testing Securely Using Data Virtualization And Data Masking

MAY 16 & 17, 2018

CLEVELAND PUBLIC AUDITORIUM,
CLEVELAND, OHIO

WWW.NEOOUG.ORG/GLOC

Who am I?

- DBA for 25+ years
 - “C” programmer prior to that
- [Co-authored six Oracle books](#)
 - Tech review on eight more
- Field services at [Delphix](#)
 - 16 years independent consultant prior to that
 - <http://EvDBT.com/>
- Married to [@DBAKevlar](#)
 - <http://DBAKevlar.com/>
 - That is our *new* home parked in front of our current home → → →
 - <http://DancesWithWinnebagos.com>



Agenda

1. Virtualization

2. Constraints

3. The Problem

4. Attempted solutions

5. Solution

6. Data masking

Virtualization

Server virtualization shares compute resources (i.e. CPU, RAM, and network), to make one physical machine into many virtual machines

Virtualization

Server virtualization shares compute resources (i.e. CPU, RAM, and network), to make one physical machine into many virtual machines

Server virtualization is the **norm** for systems administration today

Virtualization

Server virtualization shares compute resources (i.e. CPU, RAM, and network), to make one physical machine into many virtual machines

Server virtualization is the **norm** for systems administration today

This has facilitated the conversion of the data center from hardware into **scriptable software**

Virtualization

Server virtualization shares compute resources (i.e. CPU, RAM, and network), to make one physical machine into many virtual machines

Server virtualization is the **norm** for systems administration today

This has facilitated the conversion of the data center from hardware into **scriptable software**

...but it does not make **databases** or **applications** any more agile

Virtualization

Data virtualization is the rest of the virtualization solution

Virtualization

Data virtualization is the rest of the virtualization solution

If you haven't **yet** been amazed by the benefits of *server virtualization*...

...then that is because *data virtualization* has not **yet** been included.

Virtualization

Data virtualization is the rest of the virtualization solution

If you haven't **yet** been amazed by the benefits of *server virtualization*...

...then that is because *data virtualization* has not **yet** been included.

You have only seen a partial solution...

...so far

Virtualization

Stop for a moment and consider...

Q1: What *were* some of the most apparent **distortions** that arose in software development **prior** to the advent of *server virtualization*?

Virtualization

Stop for a moment and consider...

Q1: What *were* some of the most apparent **distortions** that arose in software development **prior** to the advent of *server virtualization*?

A1: Having *multiple* stacks of applications – for development, testing, training, production support, patching, etc -- on the same servers, so that they were configured and managed **NOTHING** like production

Virtualization

Stop for a moment and consider...

Q2: What *are* some of the most apparent **distortions** still present in software development **today**?

Virtualization

Stop for a moment and consider...

Q2: What *are* some of the most apparent **distortions** still present in software development **today**?

A2: Having to **share** a single DEV, SIT, QA environment because there is **not enough storage** for each project to have its own

Virtualization

Stop for a moment and consider...

Q2: What *are* some of the most apparent **distortions** still present in software development **today**?

A2: Having to **share** a single DEV, SIT, QA environment because there is **not enough storage** for each project to have its own

Even if there is enough **storage**, there isn't enough **time**.

Virtualization

New functionality and fixes must be delivered within **hours** or **days**, at most **weeks**, not months or years as in the past

Virtualization

New functionality and fixes must be delivered within **hours** or **days**, at most **weeks**, not months or years as in the past

Can such a tempo be supported with...

- a single instance of DEV
- a single instance of SIT
- a single instance of UAT

???

Virtualization

New functionality and fixes must be delivered within **hours** or **days**, at most **weeks**, not months or years as in the past

Can such a tempo be supported with...

- a single instance of DEV
- a single instance of SIT
- a single instance of UAT

???

The litmus test

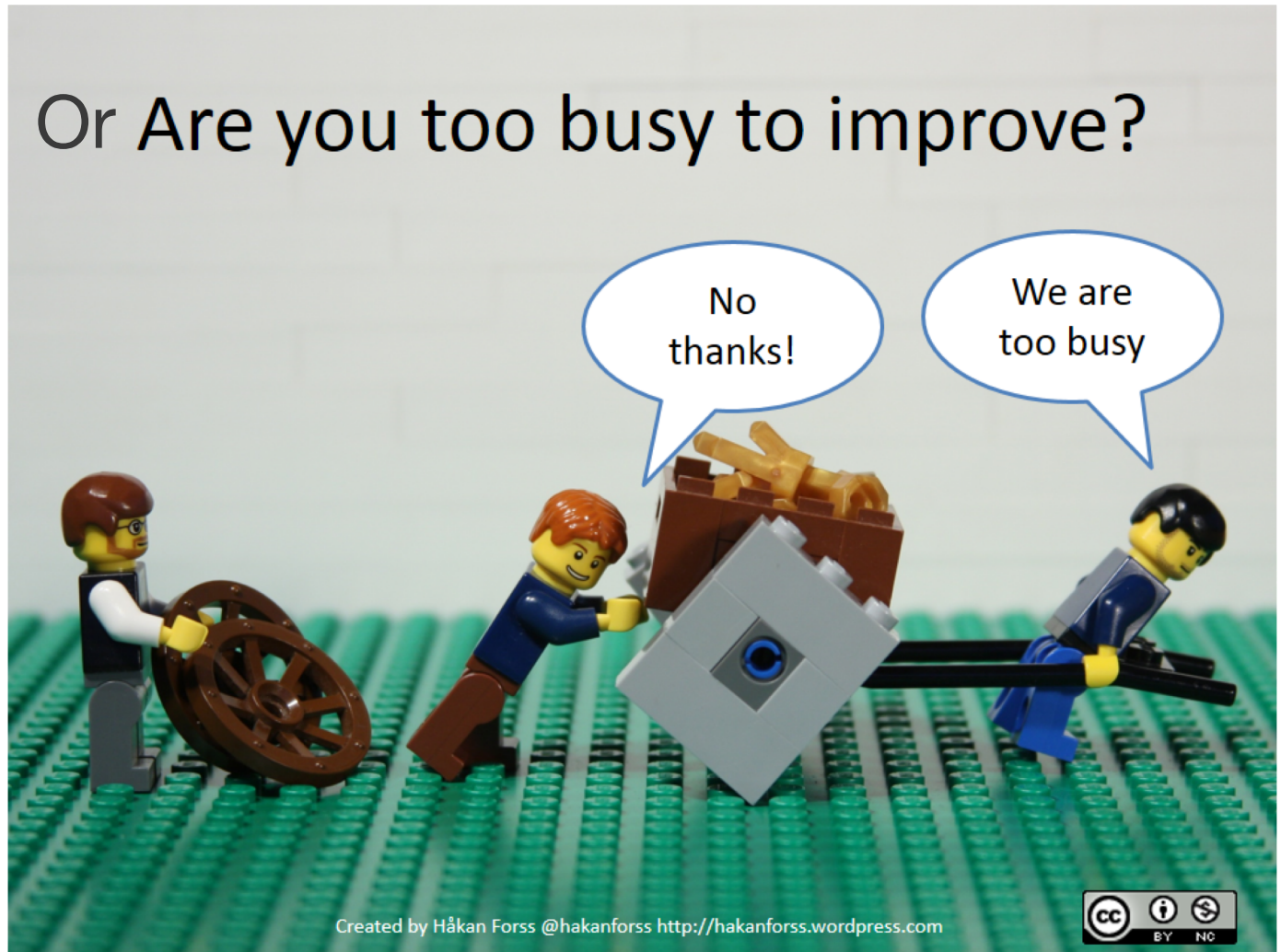
Can **your** IT infrastructure support *two-week sprints* with environments newly refreshed from production?

Virtualization

- So.....

Virtualization

- So.....
- ...are you ready to consider the next step forward?



Agenda

1. Virtualization

2. **Constraints**

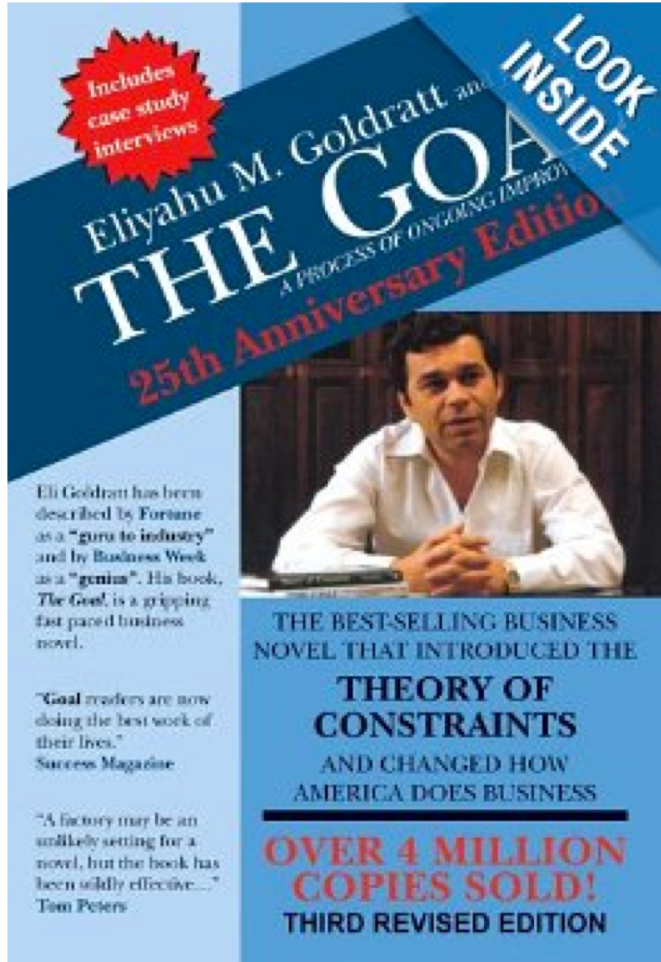
3. The Problem

4. Attempted solutions

5. Solution

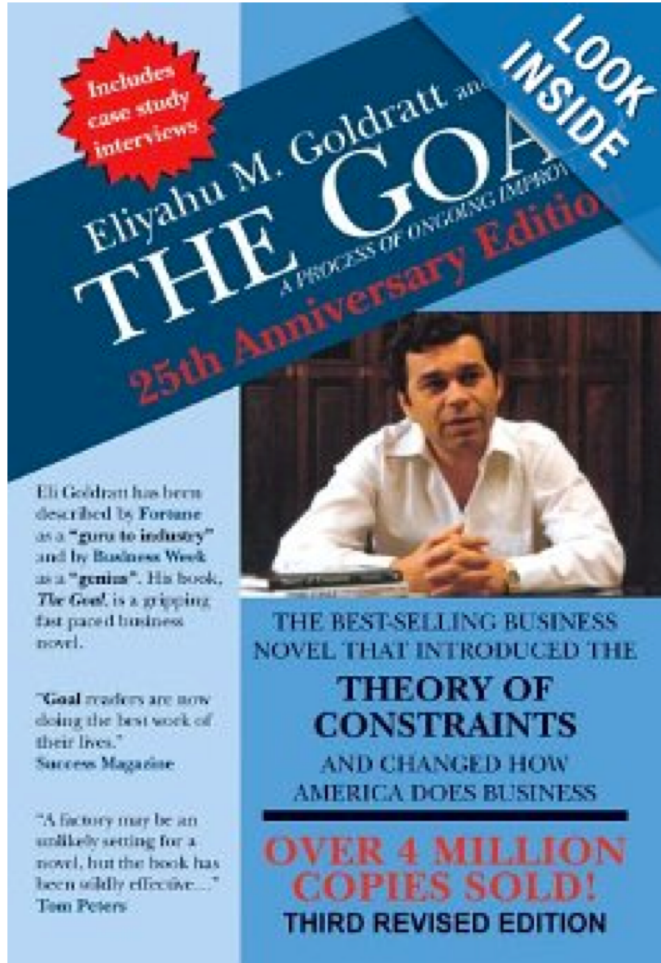
6. Data masking

The Goal: Theory Of Constraints



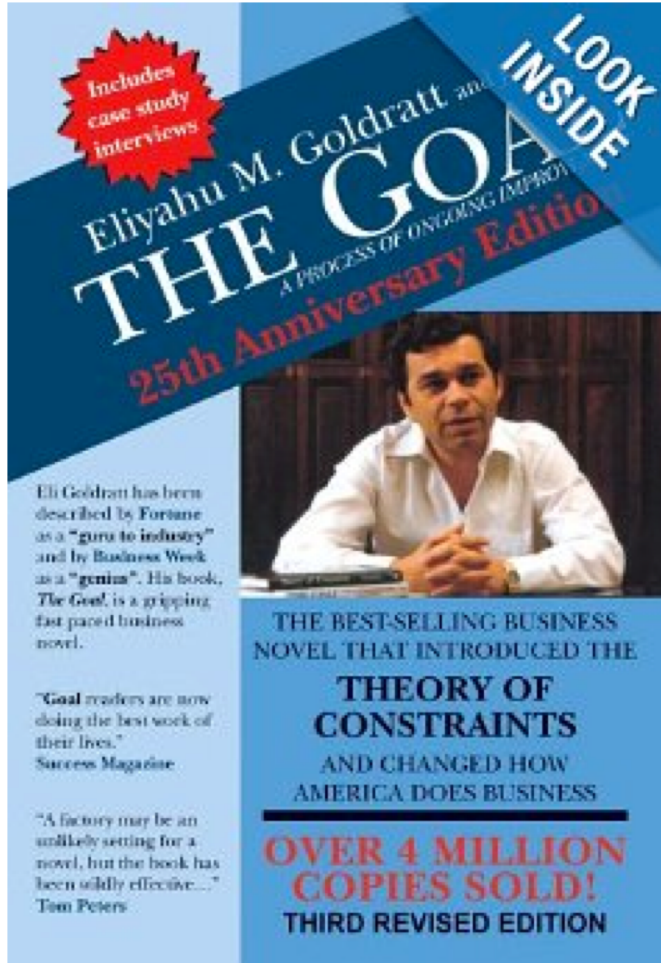
- **Long** build cycles are wasteful

The Goal: Theory Of Constraints



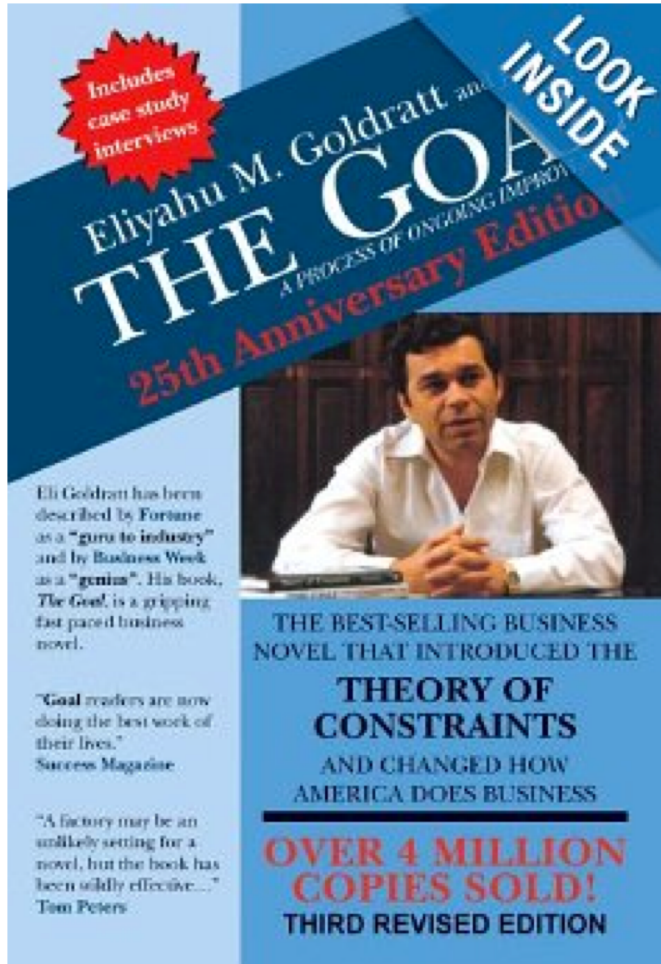
- **Long** build cycles are wasteful
- **QA/Testing** is the first victim

The Goal: Theory Of Constraints



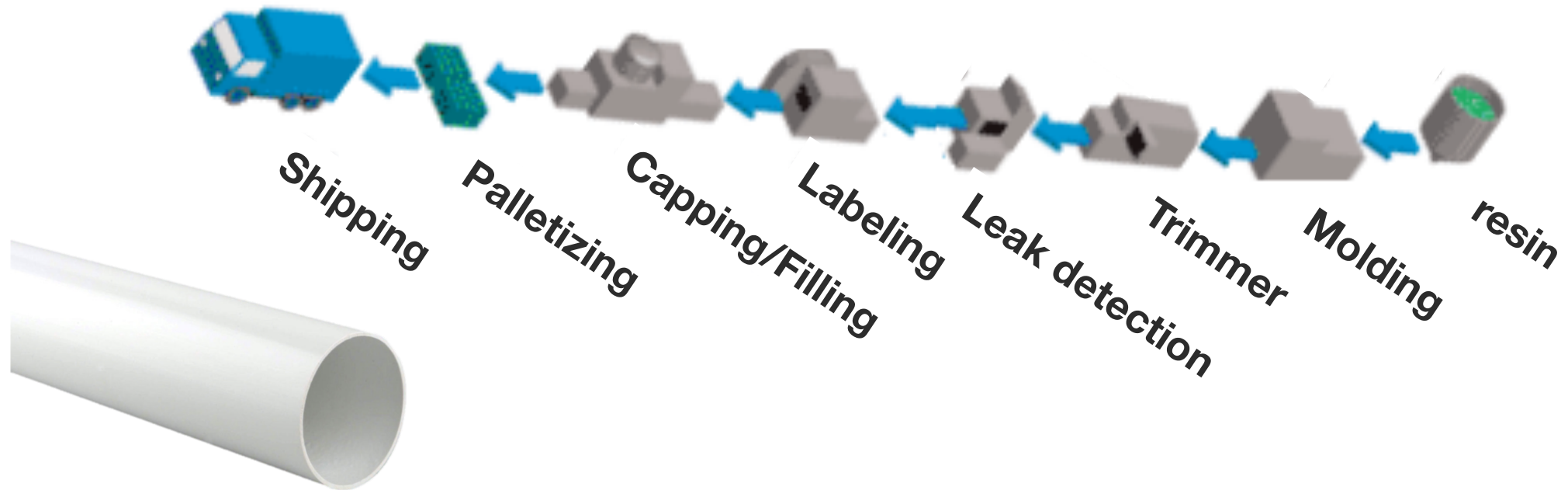
- **Long** build cycles are wasteful
- **QA/Testing** is the first victim
- **Improvement** not made at the **constraint** is an illusion

The Goal: Theory Of Constraints

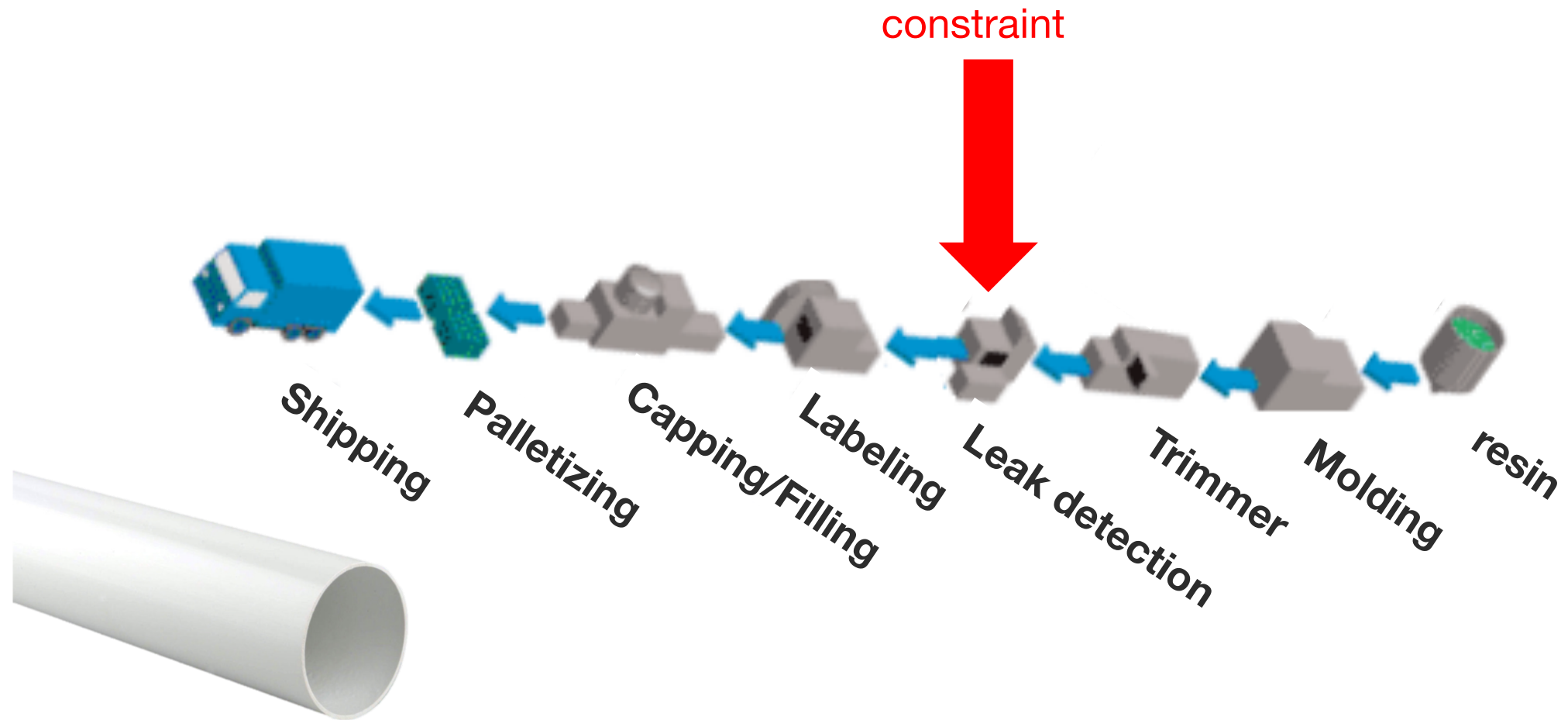


- Long build cycles are wasteful
- QA/Testing is the first victim
- Improvement not made at the **constraint** is an illusion

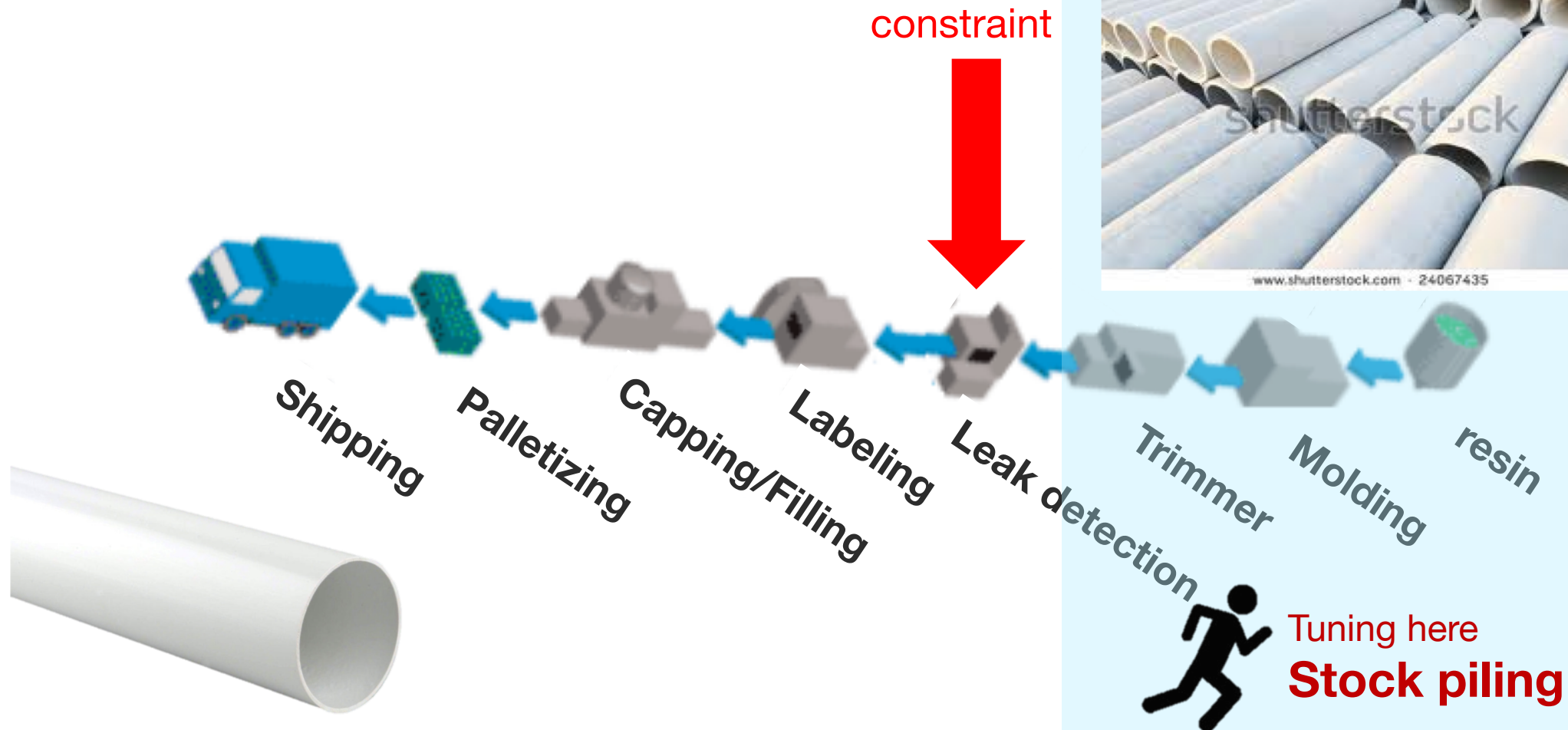
The Goal: Theory Of Constraints



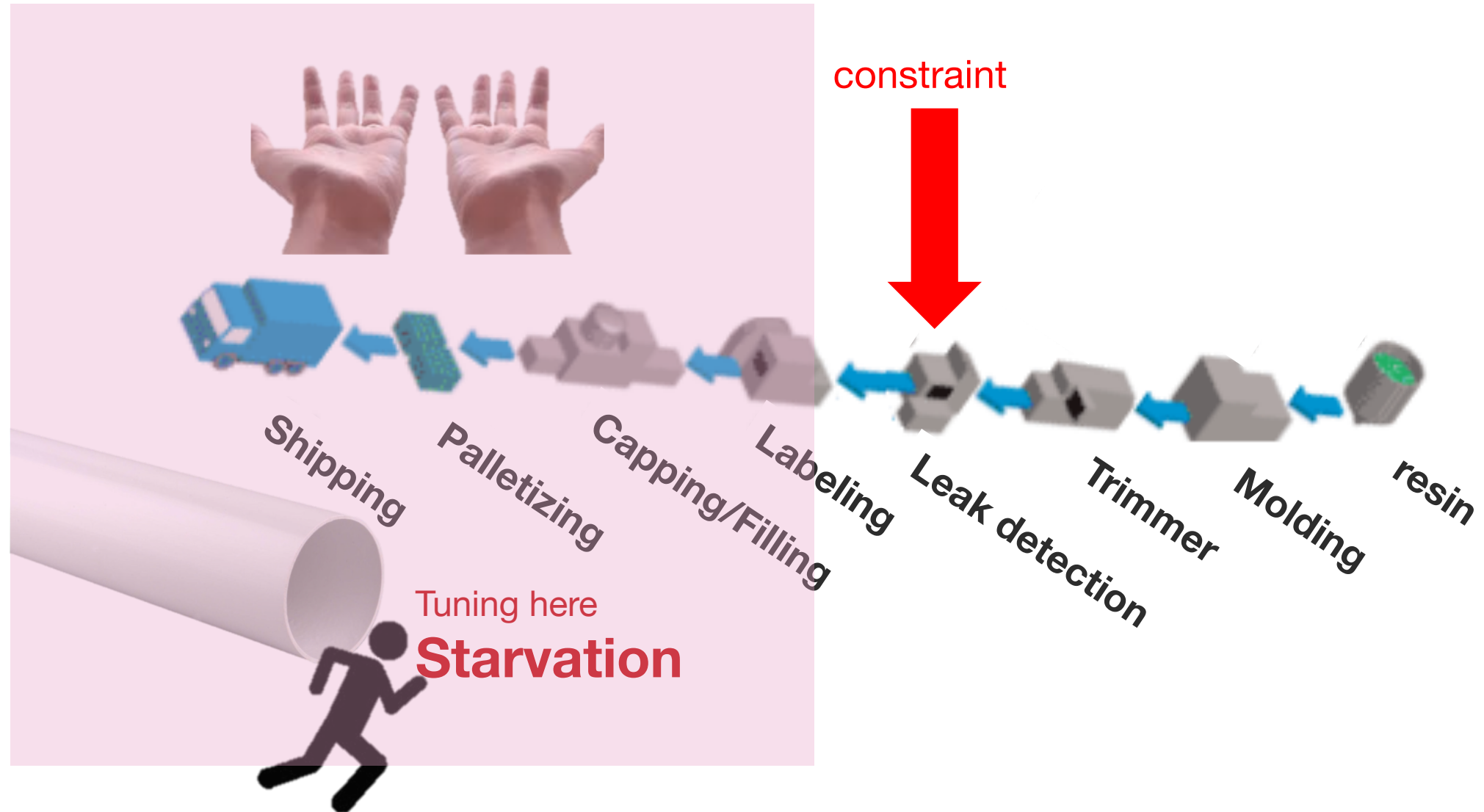
The Goal: Theory Of Constraints



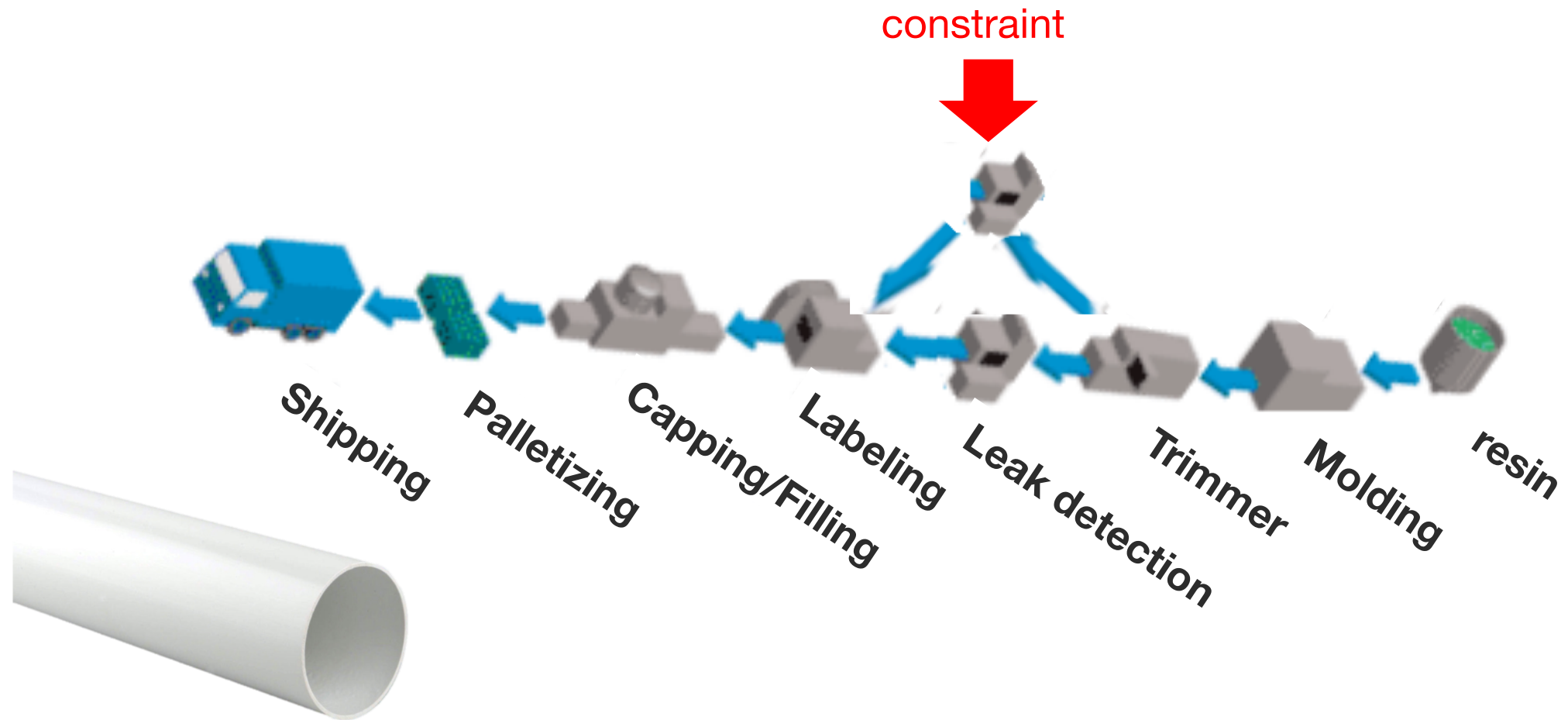
The Goal: Theory Of Constraints



The Goal: Theory Of Constraints



The Goal: Theory Of Constraints



The Phoenix Project: Theory Of Constraints For I.T.



- Goals Clarify
 - Metrics Define
 - Constraints Identify
 - Priorities Set
 - Iterations Fast
-
- CI
 - Cloud
 - Agile
 - Kanban
 - Kata

“IT is the factory floor of this century”

The Phoenix Project: Theory Of Constraints For I.T.



What is the
constraint
in IT ?

“IT is the factory floor of this century”

The Phoenix Project: Theory Of Constraints For I.T.



1. **QA setup**
2. **Dev/Build setup**
3. Code Architecture
4. Development
5. Product management

– Gene Kim, author of “*The Phoenix Project*”

“One of the most powerful things that organizations can do is to enable development and testing to get environments they need, when they need it”

“IT is the factory floor of this century”

Agenda

1. Virtualization

2. Constraints

3. **The Problem**

4. Attempted solutions

5. Solution

6. Data masking

The Problem

- PROD is what we care most about...



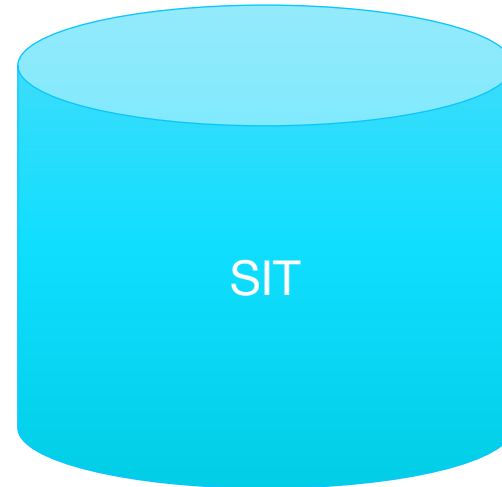
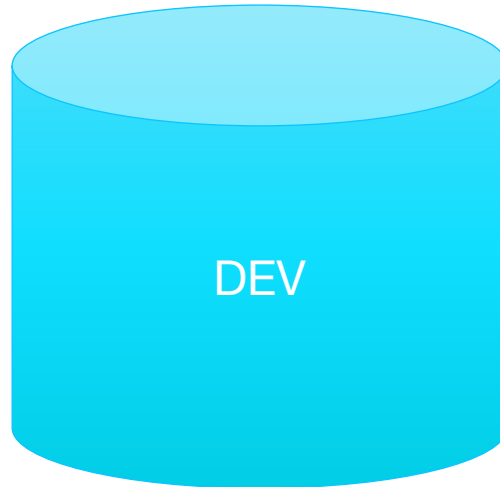
The Problem

- PROD is what we care most about...
 - But behind every PROD...
 - There is DEV...



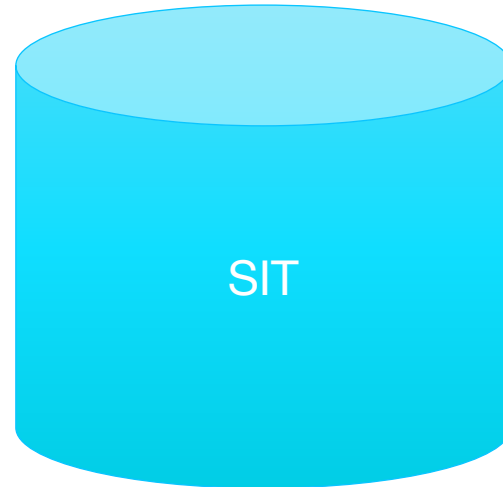
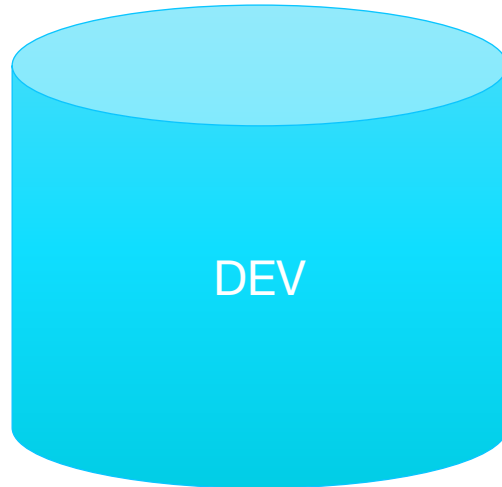
The Problem

- PROD is what we care most about...
 - But behind every PROD...
 - There is DEV, SIT...



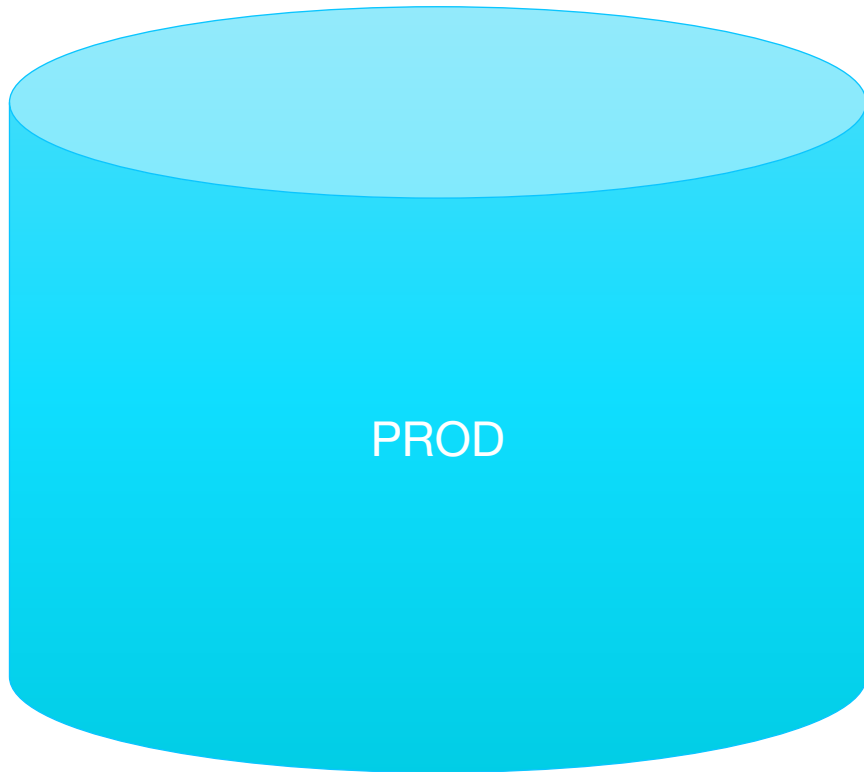
The Problem

- PROD is what we care most about...
 - But behind every PROD...
 - There is DEV, SIT, UAT...



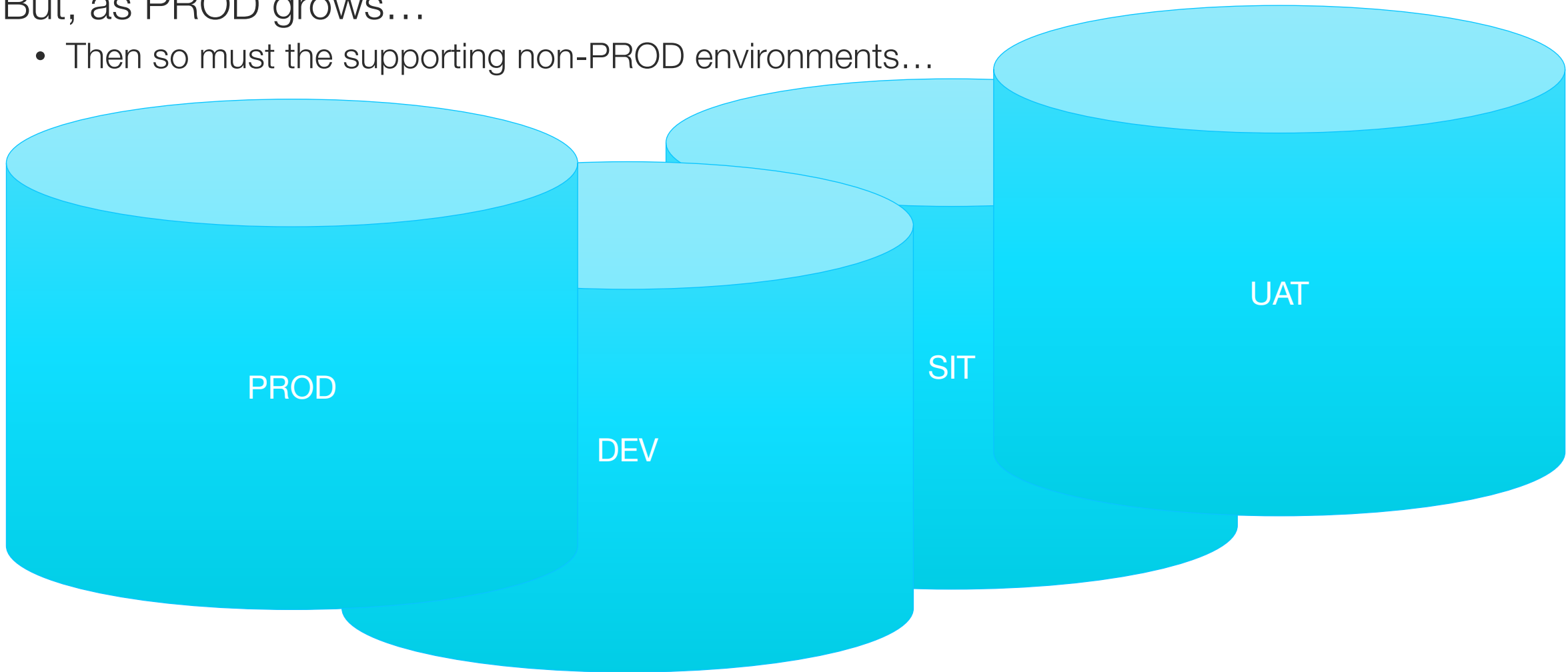
The Problem

- But, as PROD grows...



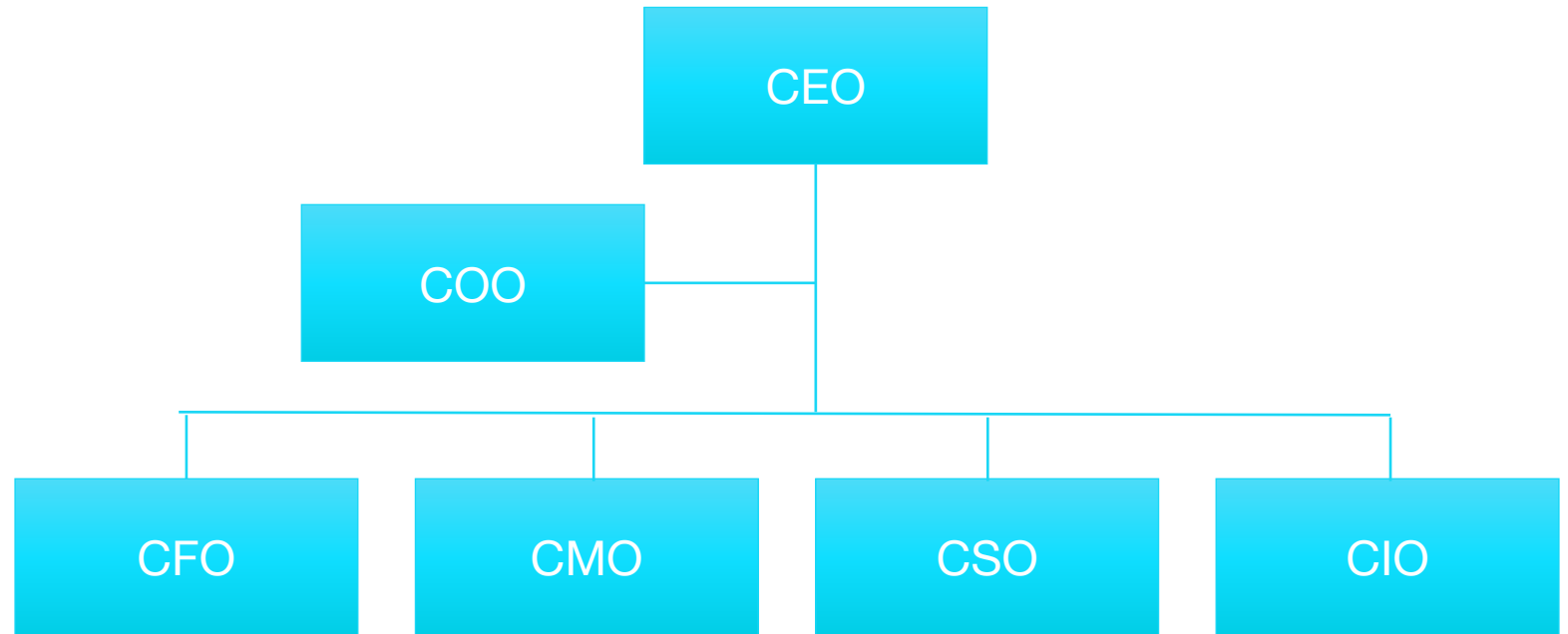
The Problem

- But, as PROD grows...
 - Then so must the supporting non-PROD environments...



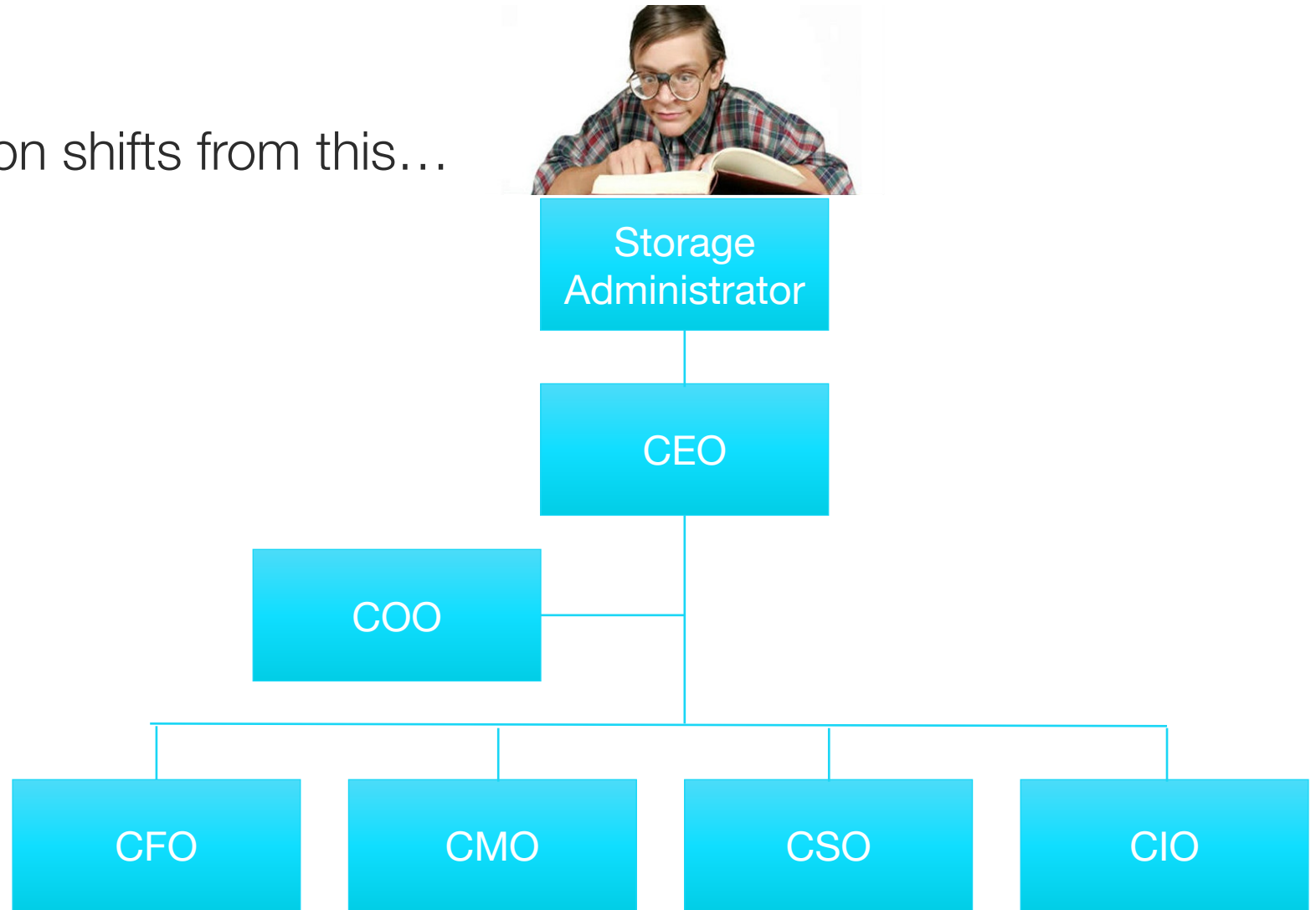
The Problem

- And so inevitably the situation shifts from this...



The Problem

- And so inevitably the situation shifts from this...
- To this...



The Problem

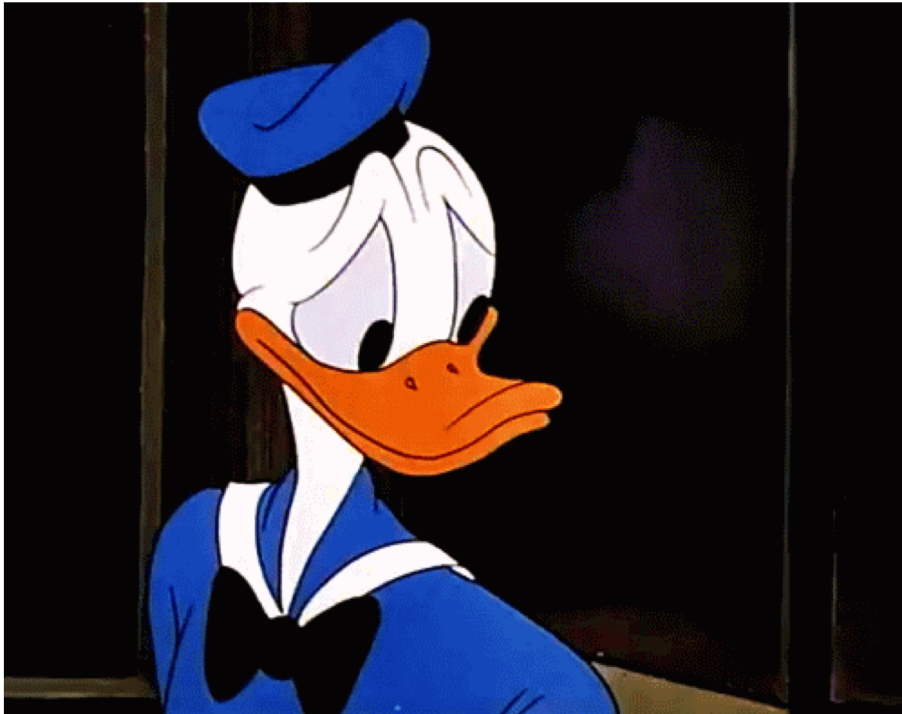
Information
technology (IT) is
generally unaware of
a problem



The Problem

We have been accommodating these limitations for decades...

Management believes it is a **fact of life**



Developers/testers have grown to accept it

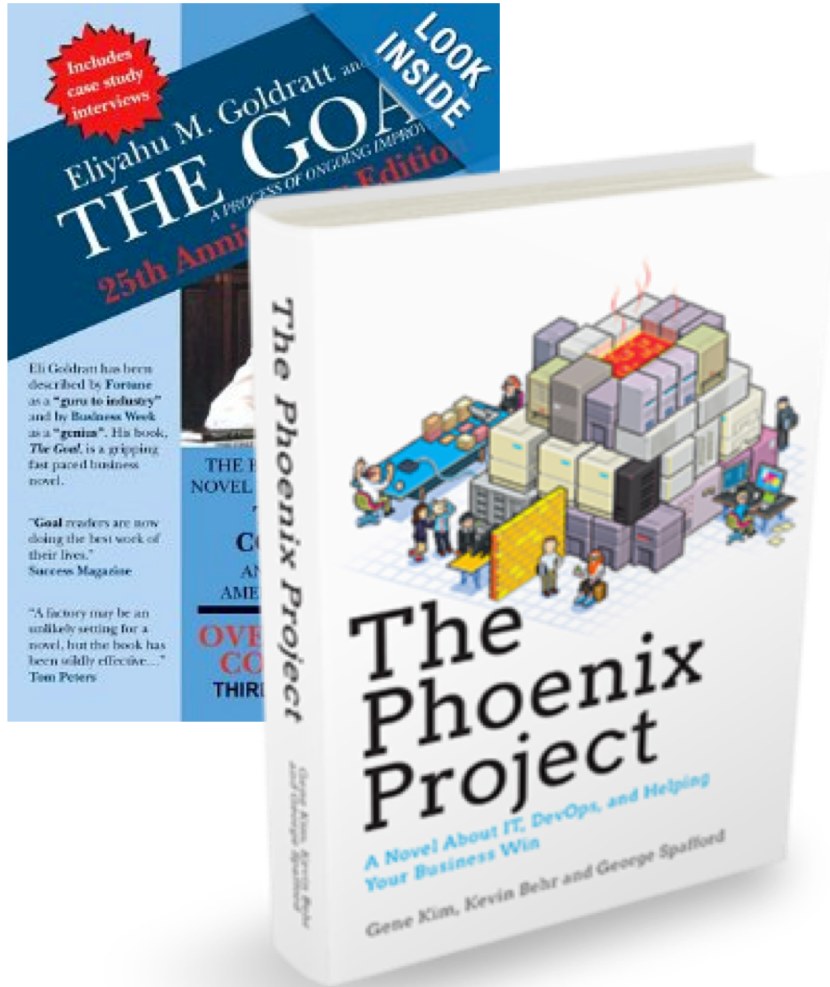


The Problem

...despite the obvious fact that the quality of delivered software is so poor



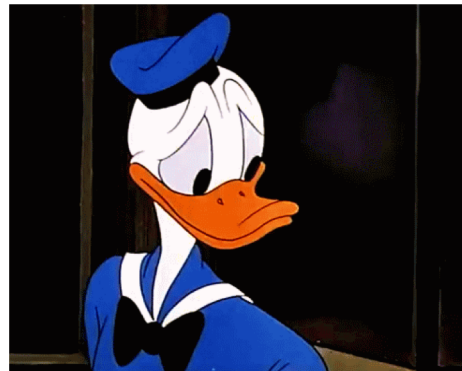
The Problem



What is the **constraint** in IT ?

“One of the most powerful things that organizations can do is to enable development and testing to get environments they need, when they need it“

– Gene Kim, author of “*The Phoenix Project*”



The Problem

- Asking again...



Agenda

1. Virtualization

2. Constraints

3. The Problem

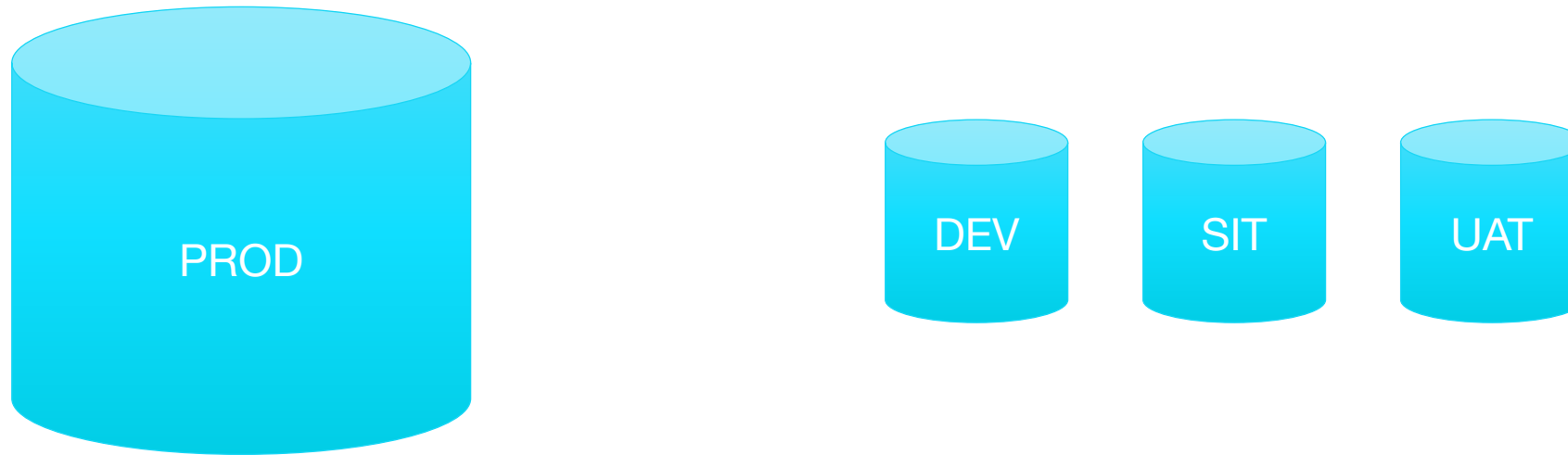
4. **Attempted solutions**

5. Solution

6. Data masking

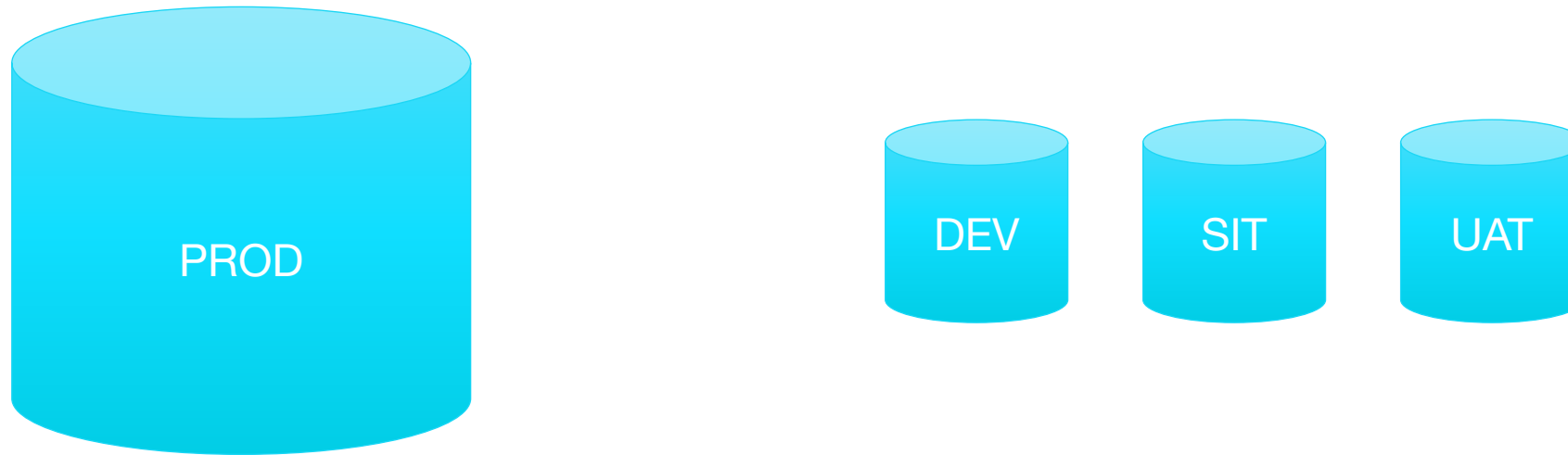
Attempted solutions

- A common “solution” is **subsetting**, so that the cloned non-production databases are consume less storage



Attempted solutions

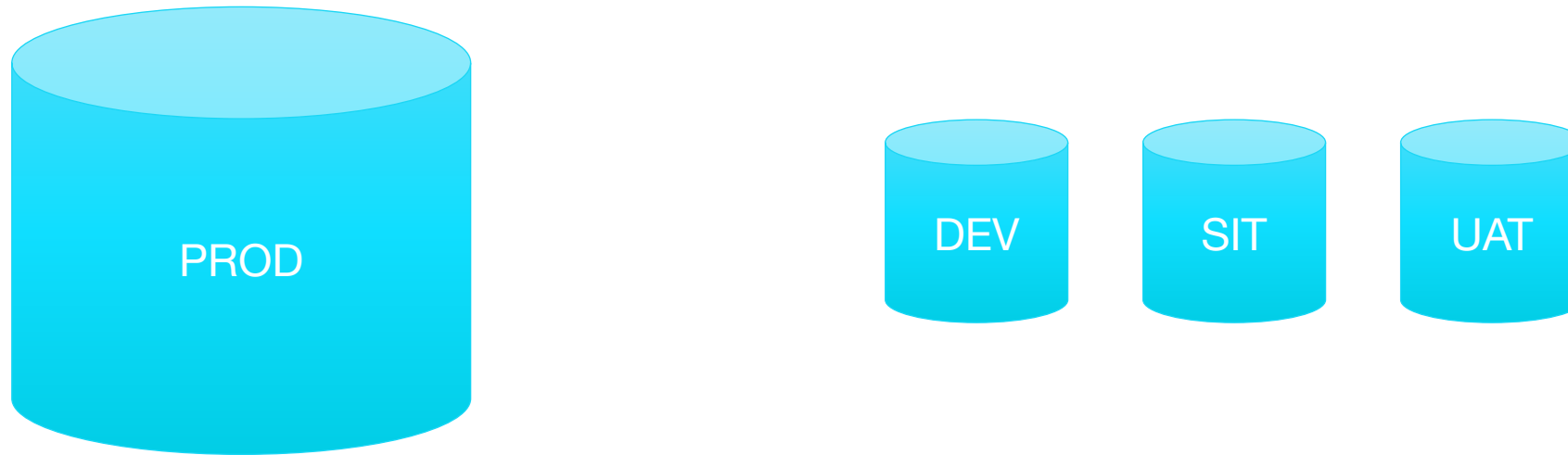
- A common “solution” is **subsetting**, so that the cloned non-production databases are consume less storage



- It is extremely difficult and complex to create a referentially-consistent subset of data

Attempted solutions

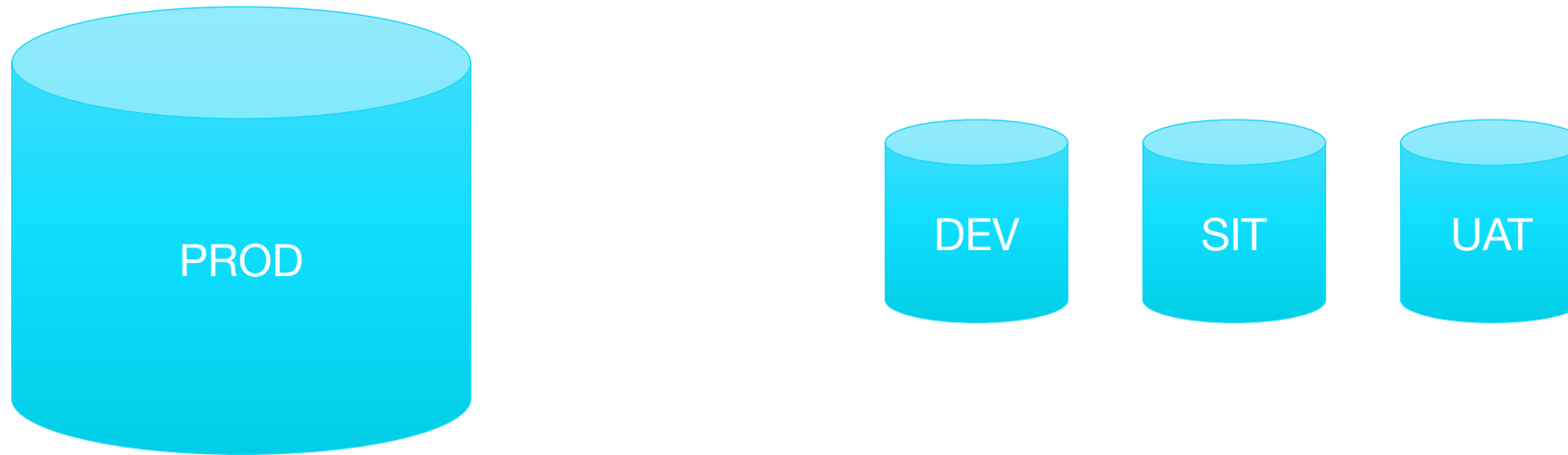
- A common “solution” is **subsetting**, so that the cloned non-production databases are consume less storage



- It is extremely difficult and complex to create a referentially-consistent subset of data
- Nonetheless, there is a thriving industry in subsetting sets of data...

Attempted solutions

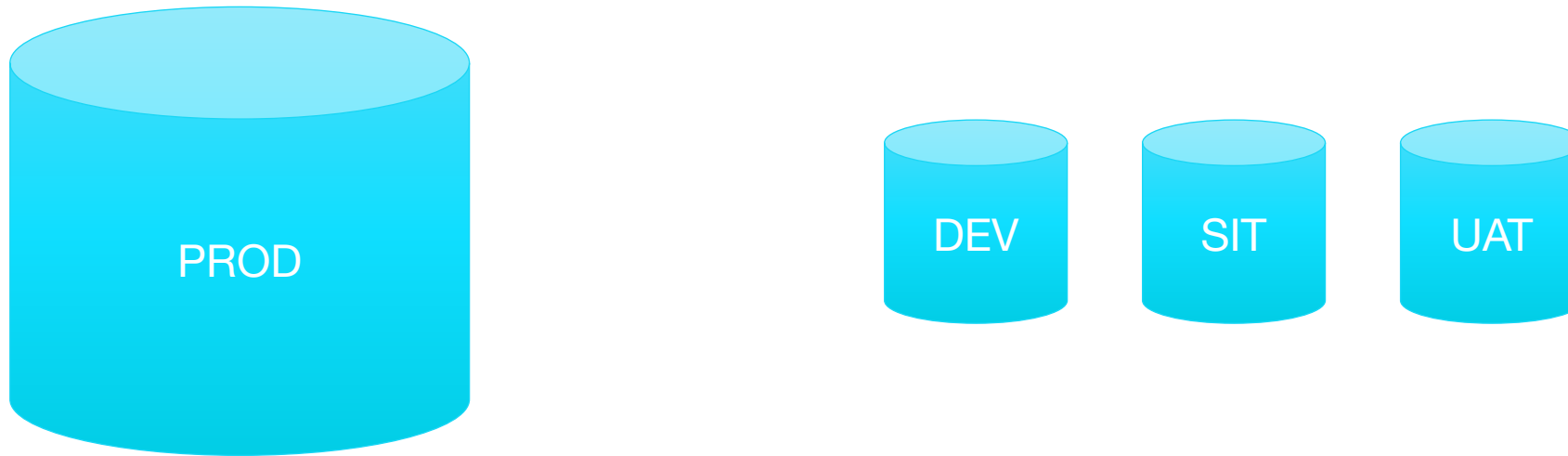
- A common “solution” is **subsetting**, so that the cloned non-production databases are consume less storage



- It is extremely difficult and complex to create a referentially-consistent subset of data
- Nonetheless, there is a thriving industry in subsetting sets of data...
 - Many insist that there are advantages to removing possible test cases and data volume

Attempted solutions

- A common “solution” is **subsetting**, so that the cloned non-production databases are consume less storage

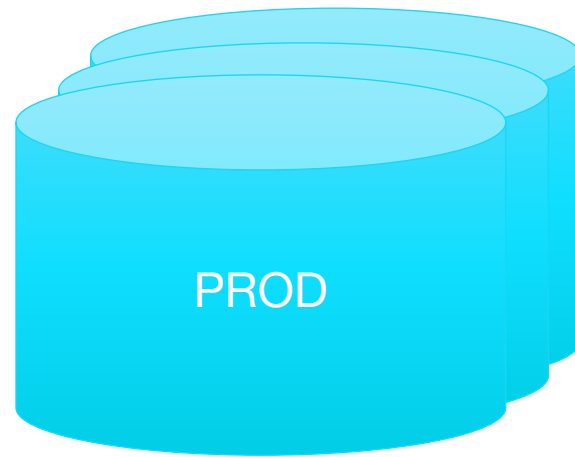


- It is extremely difficult and complex to create a referentially-consistent subset of data
- Nonetheless, there is a thriving industry in subsetting sets of data...
 - Many insist that there are advantages to removing possible test cases and data volume

Leads to buggy code...

Attempted solutions

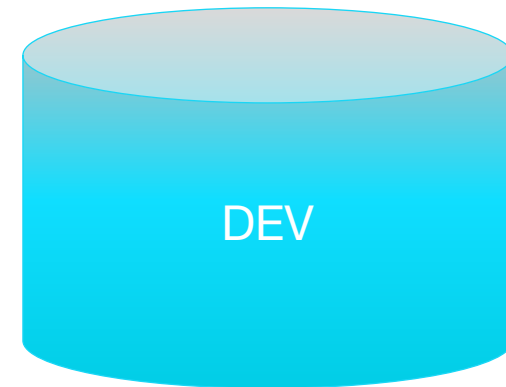
- Another common “solution” is **storage mirror splits**
 - using functionality at the storage level to quickly create clones of storage volumes



- In addition to mirroring for high-availability, create an extra mirror copy...

Attempted solutions

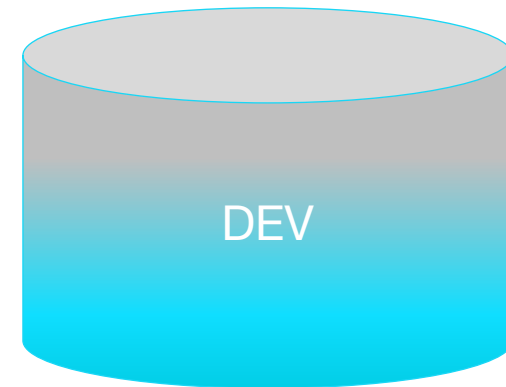
- Another common “solution” is **storage mirror splits**
 - using functionality at the storage level to quickly create clones of storage volumes



- In addition to mirroring for high-availability, create an extra mirror copy...
 - Quiesce activity to the extra mirror, and split it off to make a new volume for non-production use

Attempted solutions

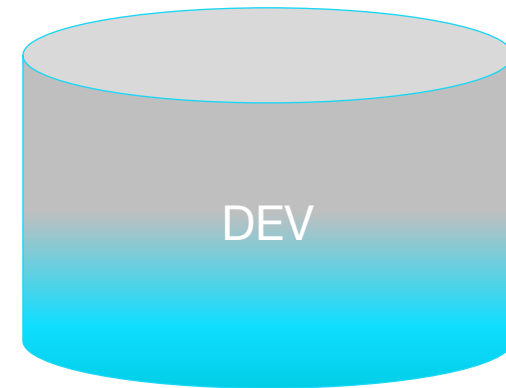
- Another common “solution” is **storage mirror splits**
 - using functionality at the storage level to quickly create clones of storage volumes



- To “refresh” the spare volume, it must rejoin the mirror group and be “resilvered”

Attempted solutions

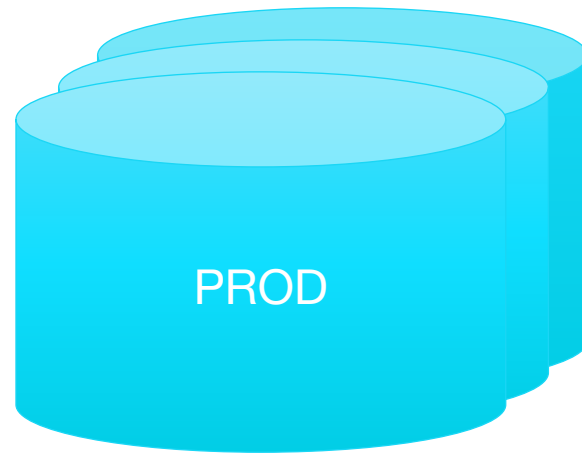
- Another common “solution” is **storage mirror splits**
 - using functionality at the storage level to quickly create clones of storage volumes



- To “refresh” the spare volume, it must rejoin the mirror group and be “resilvered”
 - Resilvering a volume is not necessarily a fast operation

Attempted solutions

- Another common “solution” is **storage mirror splits**
 - using functionality at the storage level to quickly create clones of storage volumes



- To “refresh” the spare volume, it must rejoin the mirror group and be “resilvered”
 - Resilvering a volume is not necessarily a fast operation
 - So we solved fast splits, but just once - and have not solved the problem of fast refreshes

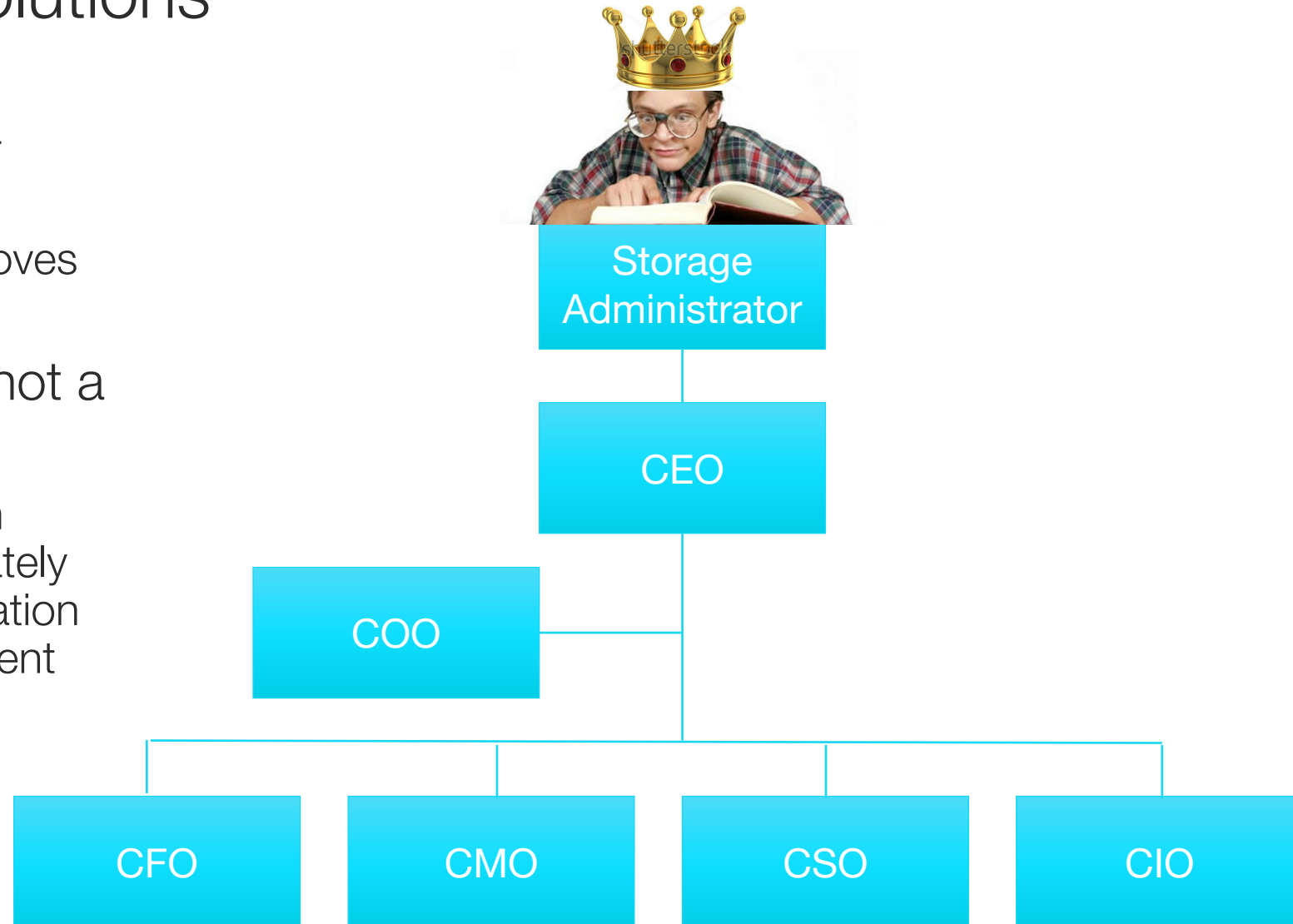
Attempted solutions

Subsetting is not a solution

- Removing data removes test credibility

Storage splits are not a solution

- Infrastructure design should not be intimately linked in with application design or management



Agenda

1. Virtualization

2. Constraints

3. The Problem

4. Attempted solutions

5. **Solution**

6. Data masking

Solution

- The solution includes...
 - Full read-write sets of data

Solution

- The solution includes...
 - Full read-write sets of data independent from their source

Solution

- The solution includes...
 - Full read-write sets of data independent from their source, from siblings

Solution

- The solution includes...
 - Full read-write sets of data independent from their source, from siblings, and from children

Solution

- The solution includes...
 - Full read-write sets of data independent from their source, from siblings, and from children
 - Provisioned and refreshed quickly

Solution

- The solution includes...
 - Full read-write sets of data independent from their source, from siblings, and from children
 - Provisioned and refreshed quickly
 - Accessible to the appropriate level of role

Solution

- The solution includes...
 - Full read-write sets of data independent from their source, from siblings, and from children
 - Provisioned and refreshed quickly
 - Accessible to the appropriate level of role
- Items for the wish-list...
 - Quickly restore to any point-in-time (a.k.a. rewind)

Solution

- The solution includes...
 - Full read-write sets of data independent from their source, from siblings, and from children
 - Provisioned and refreshed quickly
 - Accessible to the appropriate level of role
- Items for the wish-list...
 - Quickly restore to any point-in-time (a.k.a. rewind)
 - Quickly share copies with others

Solution

- The solution includes...
 - Full read-write sets of data independent from their source, from siblings, and from children
 - Provisioned and refreshed quickly
 - Accessible to the appropriate level of role
- Items for the wish-list...
 - Quickly restore to any point-in-time (a.k.a. rewind)
 - Quickly share copies with others
 - Quickly de-instantiate and store copies, as well as quickly retrieve and re-instantiate

Solution

- The solution includes...
 - Full read-write sets of data independent from their source, from siblings, and from children
 - Provisioned and refreshed quickly
 - Accessible to the appropriate level of role
- Items for the wish-list...
 - Quickly restore to any point-in-time (a.k.a. rewind)
 - Quickly share copies with others
 - Quickly de-instantiate and store copies, as well as quickly retrieve and re-instantiate
- **Thought exercise...**
 - If cost and physics were not constraints, would you provide each developer and tester with as many environments as they needed, when they needed them?

Solution

- The solution includes...
 - Full read-write sets of data independent from their source, from siblings, and from children
 - Provisioned and refreshed quickly
 - Accessible to the appropriate level of role
- Items for the wish-list...
 - Quickly restore to any point-in-time (a.k.a. rewind)
 - Quickly share copies with others
 - Quickly de-instantiate and store copies, as well as quickly retrieve and re-instantiate
- **Thought exercise...**
 - If cost and physics were not constraints, would you provide each developer and tester with as many environments as they needed, when they needed them?
 - Or would you still have them serialize their work within shared development and testing environments?

Solution

- The biggest problem with provisioning environments and databases is all the duplication

Solution

- The biggest problem with provisioning environments and databases is all the duplication
 - Most (if not all) of the blocks comprising the database or file-system are **exactly** the same

Solution

- The biggest problem with provisioning environments and databases is all the duplication
 - Most (if not all) of the blocks comprising the database or file-system are **exactly** the same
- What if we store only one copy of each version of a data block?

Solution

- The biggest problem with provisioning environments and databases is all the duplication
 - Most (if not all) of the blocks comprising the database or file-system are **exactly** the same
- What if we store only one copy of each version of a data block?
 - Copy-on-write (COW)
 - Never modify a data block, but insert a new version instead

Solution

- The biggest problem with provisioning environments and databases is all the duplication
 - Most (if not all) of the blocks comprising the database or file-system are **exactly** the same
- What if we store only one copy of each version of a data block?
 - Copy-on-write (COW)
 - Never modify a data block, but insert a new version instead
- Then let the storage figure out what we're supposed to be seeing?

Solution

- The biggest problem with provisioning environments and databases is all the duplication
 - Most (if not all) of the blocks comprising the database or file-system are **exactly** the same
- What if we store only one copy of each version of a data block?
 - Copy-on-write (COW)
 - Never modify a data block, but insert a new version instead
- Then let the storage figure out what we're supposed to be seeing?
 - The storage maintains the proper relationships and versions on retrieval

Solution

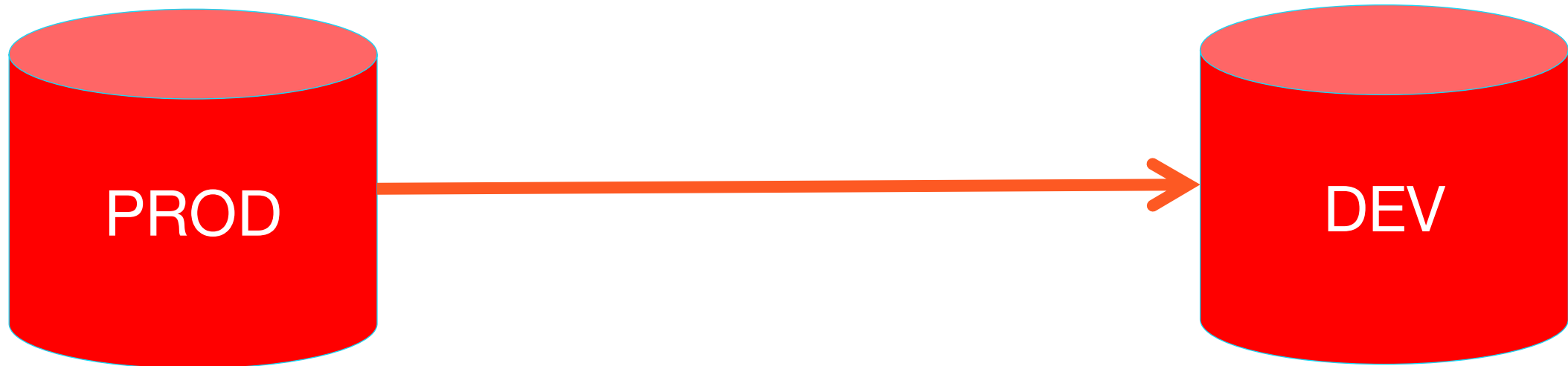
- The biggest problem with provisioning environments and databases is all the duplication
 - Most (if not all) of the blocks comprising the database or file-system are **exactly** the same
- What if we store only one copy of each version of a data block?
 - Copy-on-write (COW)
 - Never modify a data block, but insert a new version instead
- Then let the storage figure out what we're supposed to be seeing?
 - The storage maintains the proper relationships and versions on retrieval
- And then make it portable so that ANY platform (Unix/Linux, Windows) can use it?

Solution

- The biggest problem with provisioning environments and databases is all the duplication
 - Most (if not all) of the blocks comprising the database or file-system are **exactly** the same
- What if we store only one copy of each version of a data block?
 - Copy-on-write (COW)
 - Never modify a data block, but insert a new version instead
- Then let the storage figure out what we're supposed to be seeing?
 - The storage maintains the proper relationships and versions on retrieval
- And then make it portable so that ANY platform (Unix/Linux, Windows) can use it?
 - NFS (network file system – Unix/Linux), iSCSI (Windows)

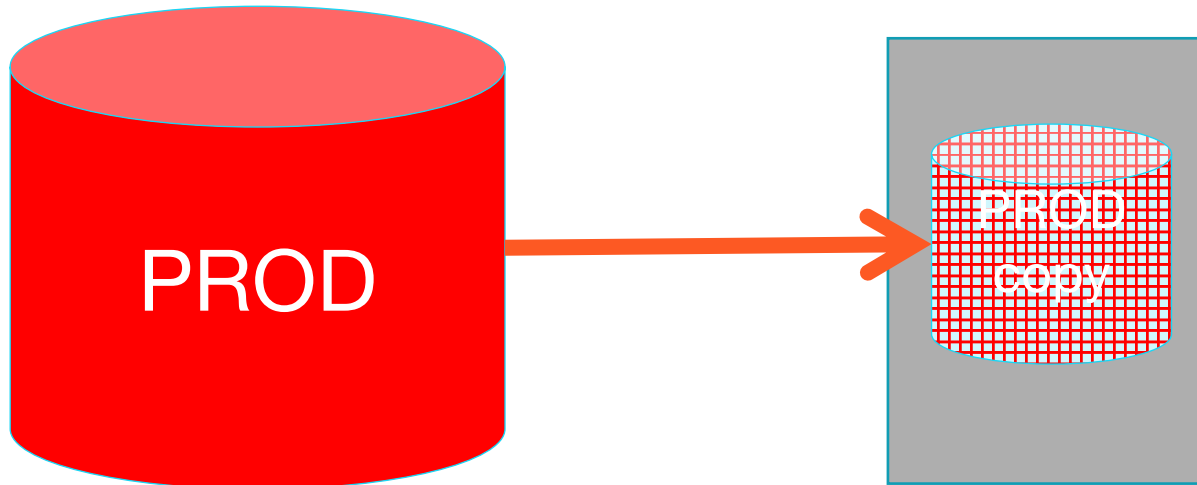
Solution

- Cloning data the old way...



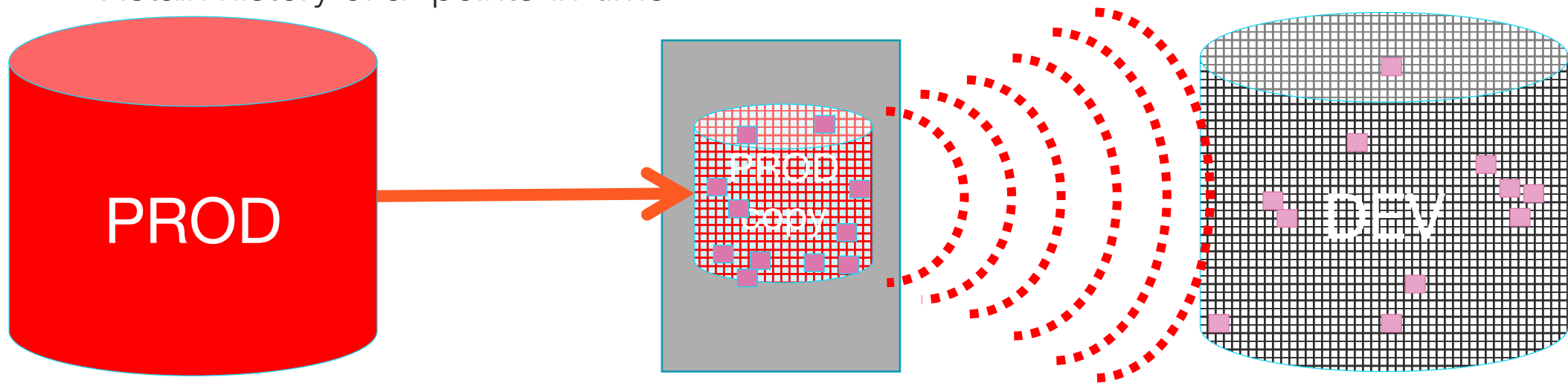
Solution

- Cloning data the smart way...
 - Capture changes incrementally forever
 - Retain history of all points-in-time



Solution

- Cloning data the smart way...
 - Capture changes incrementally forever...
 - Retain history of all points-in-time



- Thin-provision a virtual dataset from a specified point-in-time
- Project the virtual dataset over network-attached storage to the target server
- Store only changes to the virtual dataset

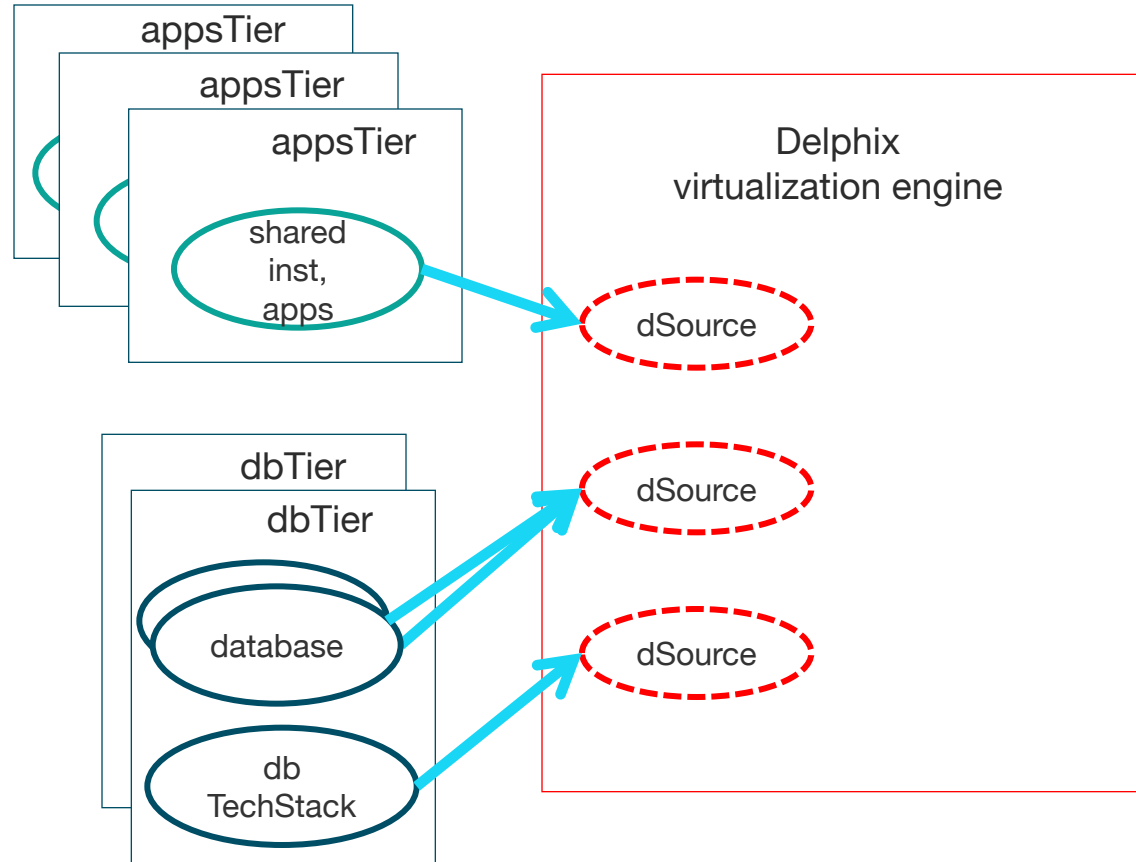
Solution

- Data blocks are data blocks
 - Whether they belong to a database, or they are just files within directories
 - Oracle, SQL Server, SAP ASE (Sybase), SAP HANA, DB2
 - Data virtualization operates at the data-block level
 - Capturing changes from sources
 - Projecting virtual dataset copies across network-attached storage
 - Retaining changes to virtual datasets

Solution

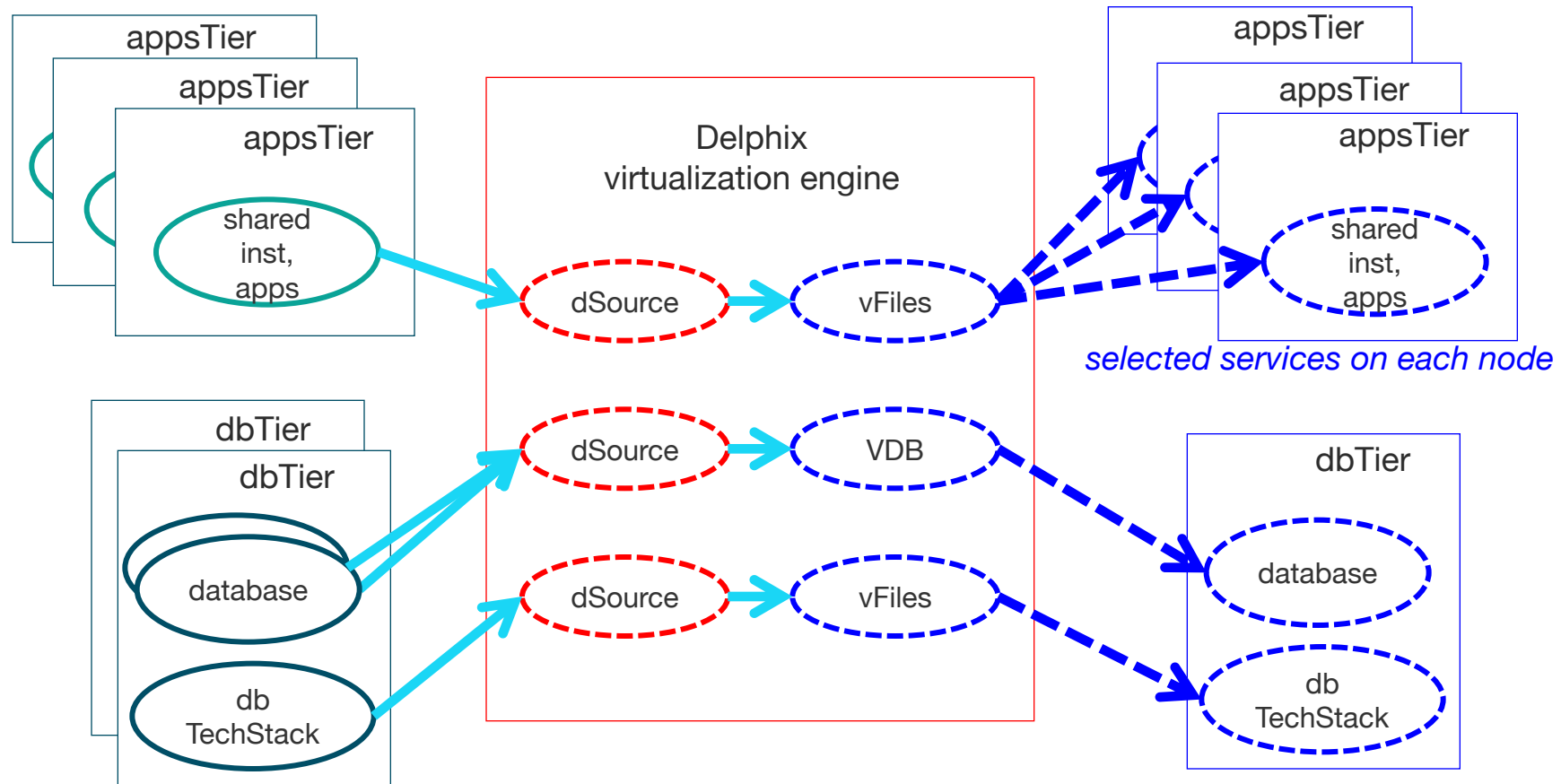
- Data blocks are data blocks
 - Whether they belong to a database, or they are just files within directories
 - Oracle, SQL Server, SAP ASE (Sybase), SAP HANA, DB2
 - Data virtualization operates at the data-block level
 - Capturing changes from sources
 - Projecting virtual dataset copies across network-attached storage
 - Retaining changes to virtual datasets
- Blocks are objects, it is how you apply methods to objects that is important
 - Managing blocks belonging to a database is different from managing blocks belonging to flat-files
 - Storage management must be integrated beneath higher-level application management
 - The user of data virtualization is not a storage administrator, a systems administrator, or a virtualization administrator
 - it is a database administrator, a developer, or a tester

Solution



- **Real-life example:**
Oracle E-Business Suites
 - Full-stack application consisting of two tiers:
 1. dbTier
 2. appsTier
 - Each tier can consist of multiple nodes:
 - dbTier can scale using database clustering
 - appsTier can scale by adding nodes for specific services

Solution



Solution

- Data virtualization is...

Solution

- Data virtualization is...
 - ...database only, or databases with full-stack applications
 - Databases supported: Oracle, SQL Server, SAP ASE (Sybase), SAP HANA, DB2
 - Custom with “hooks”: any application software

Solution

- Data virtualization is...
 - ...database only, or databases with full-stack applications
 - Databases supported: Oracle, SQL Server, SAP ASE (Sybase), SAP HANA, DB2
 - Custom with “hooks”: any application software
 - ...as many copies as you **really** need
 - Why doesn't every tester and developer have their own private environments today?

Solution

- Data virtualization is...
 - ...database only, or databases with full-stack applications
 - Databases supported: Oracle, SQL Server, SAP ASE (Sybase), SAP HANA, DB2
 - Custom with “hooks”: any application software
 - ...as many copies as you **really** need
 - Why doesn't every tester and developer have their own private environments today?
 - ...full size
 - Subsetting is an awful solution

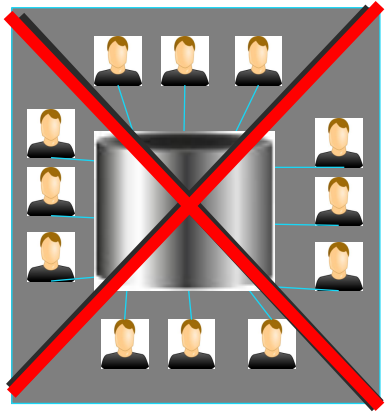
Solution

- Data virtualization is...
 - ...database only, or databases with full-stack applications
 - Databases supported: Oracle, SQL Server, SAP ASE (Sybase), SAP HANA, DB2
 - Custom with “hooks”: any application software
 - ...as many copies as you **really** need
 - Why doesn't every tester and developer have their own private environments today?
 - ...full size
 - Subsetting is an awful solution
 - ...self Service
 - It should not be necessary for testers and developers to involve the infrastructure team to manage virtual copies

Solution

- Data virtualization is...
 - ...database only, or databases with full-stack applications
 - Databases supported: Oracle, SQL Server, SAP ASE (Sybase), SAP HANA, DB2
 - Custom with “hooks”: any application software
 - ...as many copies as you **really** need
 - Why doesn't every tester and developer have their own private environments today?
 - ...full size
 - Subsetting is an awful solution
 - ...self Service
 - It should not be necessary for testers and developers to involve the infrastructure team to manage virtual copies
 - ...automatable
 - Graphical interface
 - SSH-based command-line interface (CLI) for automation
 - Web-based REST API for integration

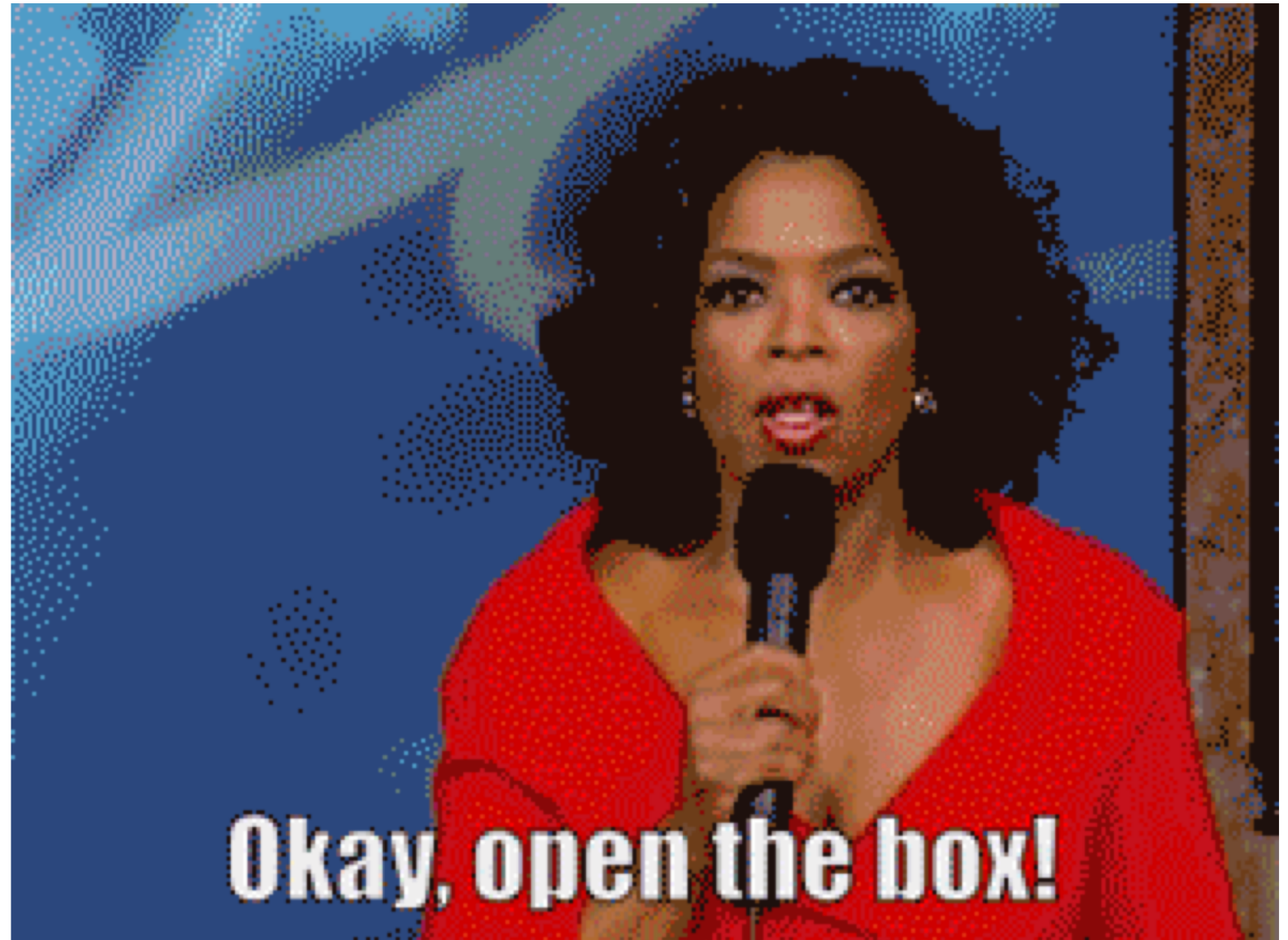
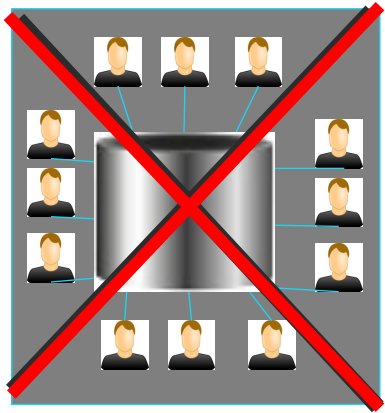
Solution



Instead of many people huddling around a single copy of a database or application

GIF by Steve Karam

Solution



GIF by Steve Karam

Agenda

1. Virtualization

2. Constraints

3. The Problem

4. Attempted solutions

5. Solution

6. **Data masking**

Data masking

- So...

Data masking

- So...
 - Data virtualization makes it fast and easy to create virtual copies from production in development, testing, training, production support, and other non-production environments

Data masking

- So...
 - Data virtualization makes it fast and easy to create virtual copies from production in development, testing, training, production support, and other non-production environments
- As a result...

Data masking

- So...
 - Data virtualization makes it fast and easy to create virtual copies from production in development, testing, training, production support, and other non-production environments
- As a result...

Somewhere

Data masking

- So...
 - Data virtualization makes it fast and easy to create virtual copies from production in development, testing, training, production support, and other non-production environments
- As a result...

Somewhere
there is a security administrator
screaming in torment and frustration

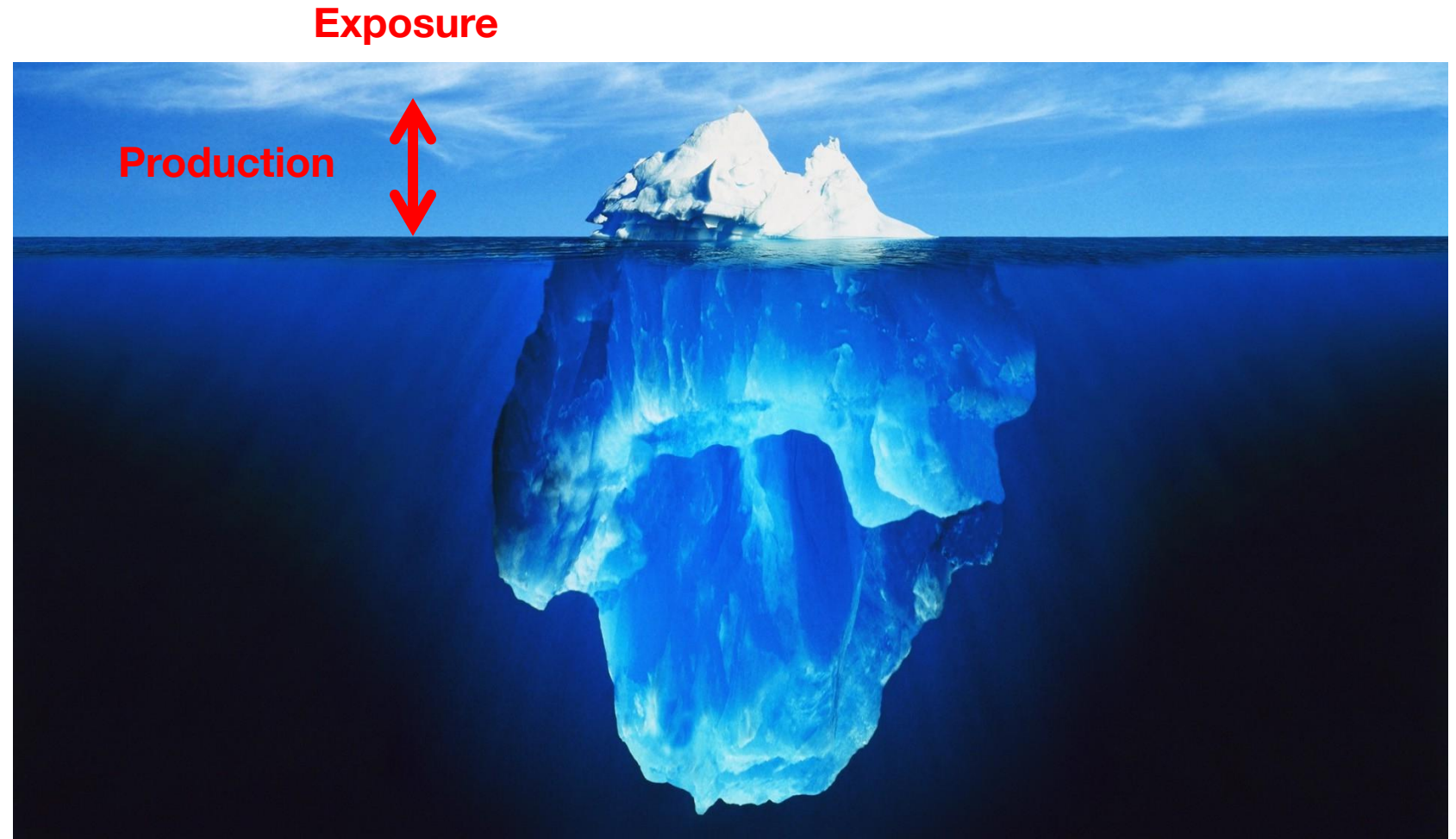
Data masking

- So...
 - Data virtualization makes it fast and easy to create virtual copies from production in development, testing, training, production support, and other non-production environments
- As a result...

Somewhere
there is a security administrator
screaming in torment and frustration
and with very good reason!

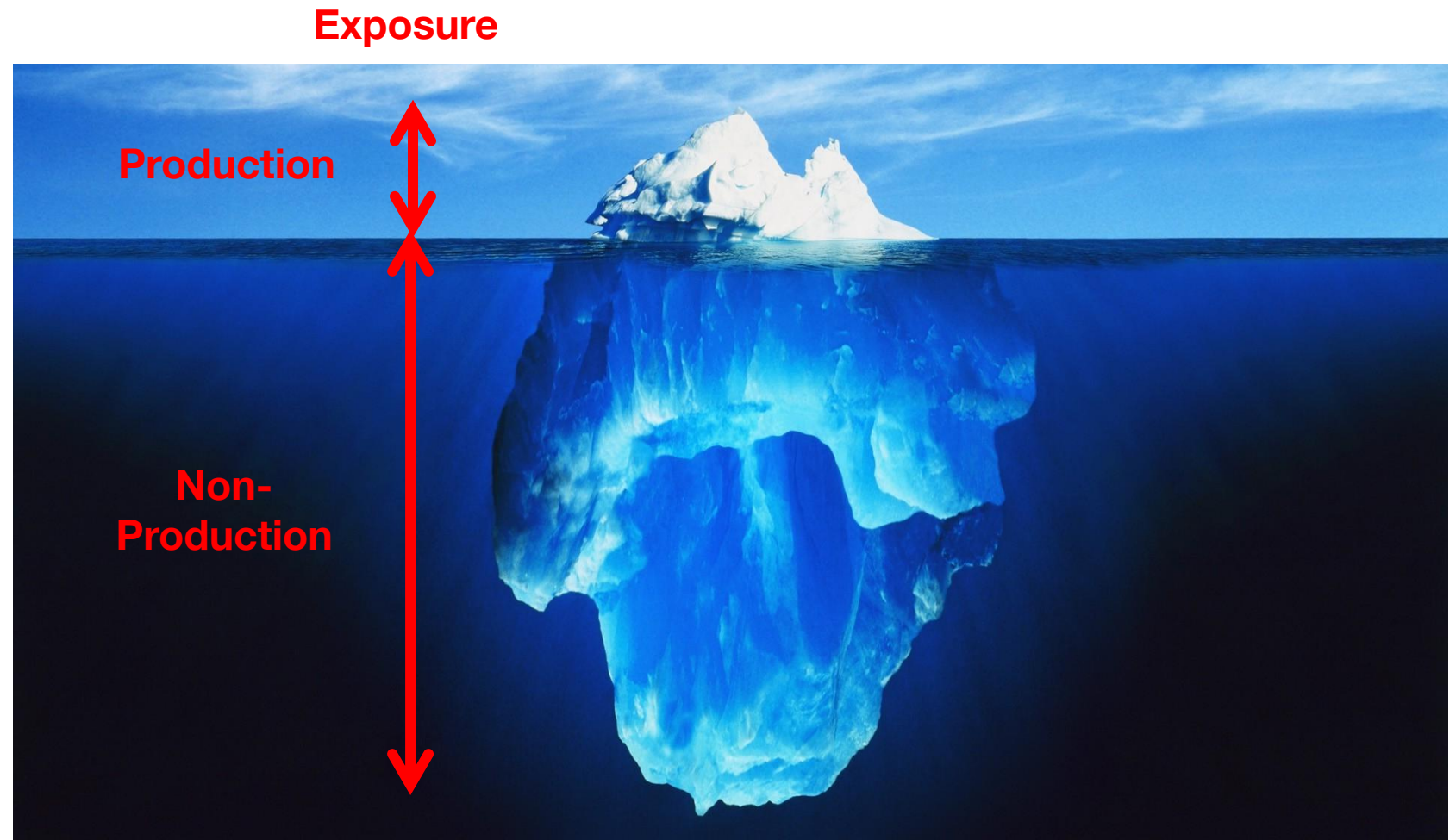
Data masking

- Non-production environments represent an enormous increase in the *surface area of risk* for exposure of sensitive production data



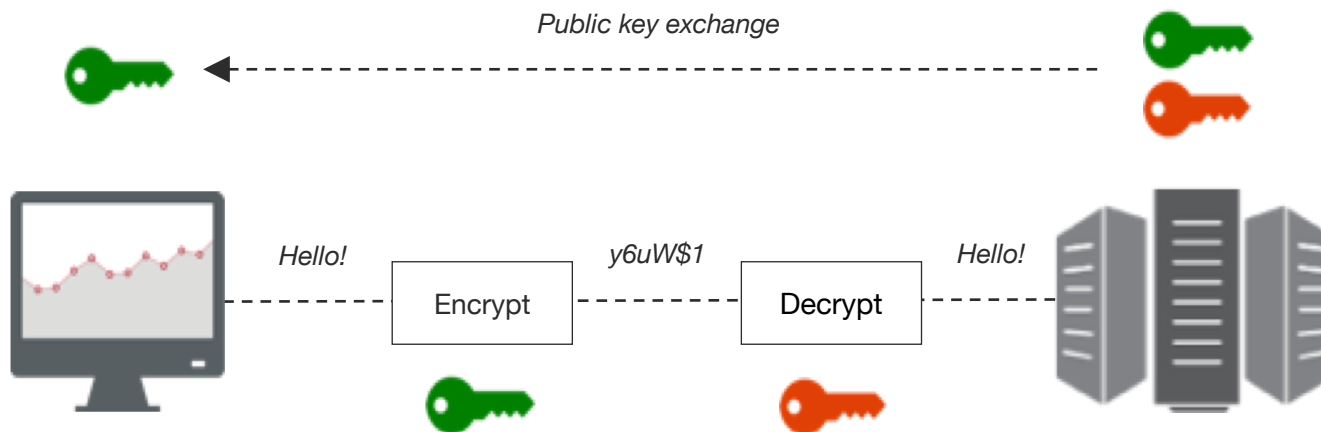
Data masking

- Non-production environments represent an enormous increase in the *surface area of risk* for exposure of sensitive production data



Data masking

- **Encryption** is the process of encoding data in such a way that only authenticated and authorized parties can decrypt it
- *Reversible* obfuscation



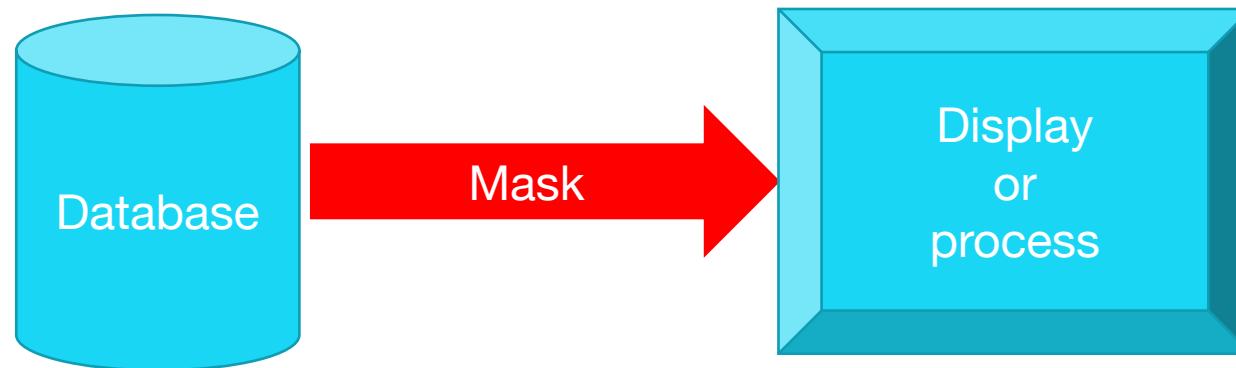
ADVANTAGES

- ▶ Effective for sending data such as emails or files between two parties (*data in-flight*)
- ▶ Effective for protecting data in a production application (*data at-rest*)

- In non-production, developers and testers must be authorized to decrypt data to do their jobs
- What if they aren't really authorized to view sensitive data?

Data Masking

- **Masking data in-flight** is the obfuscation of data **after** it has been retrieved from storage **at-rest**
- Masking = ***non-reversible*** obfuscation



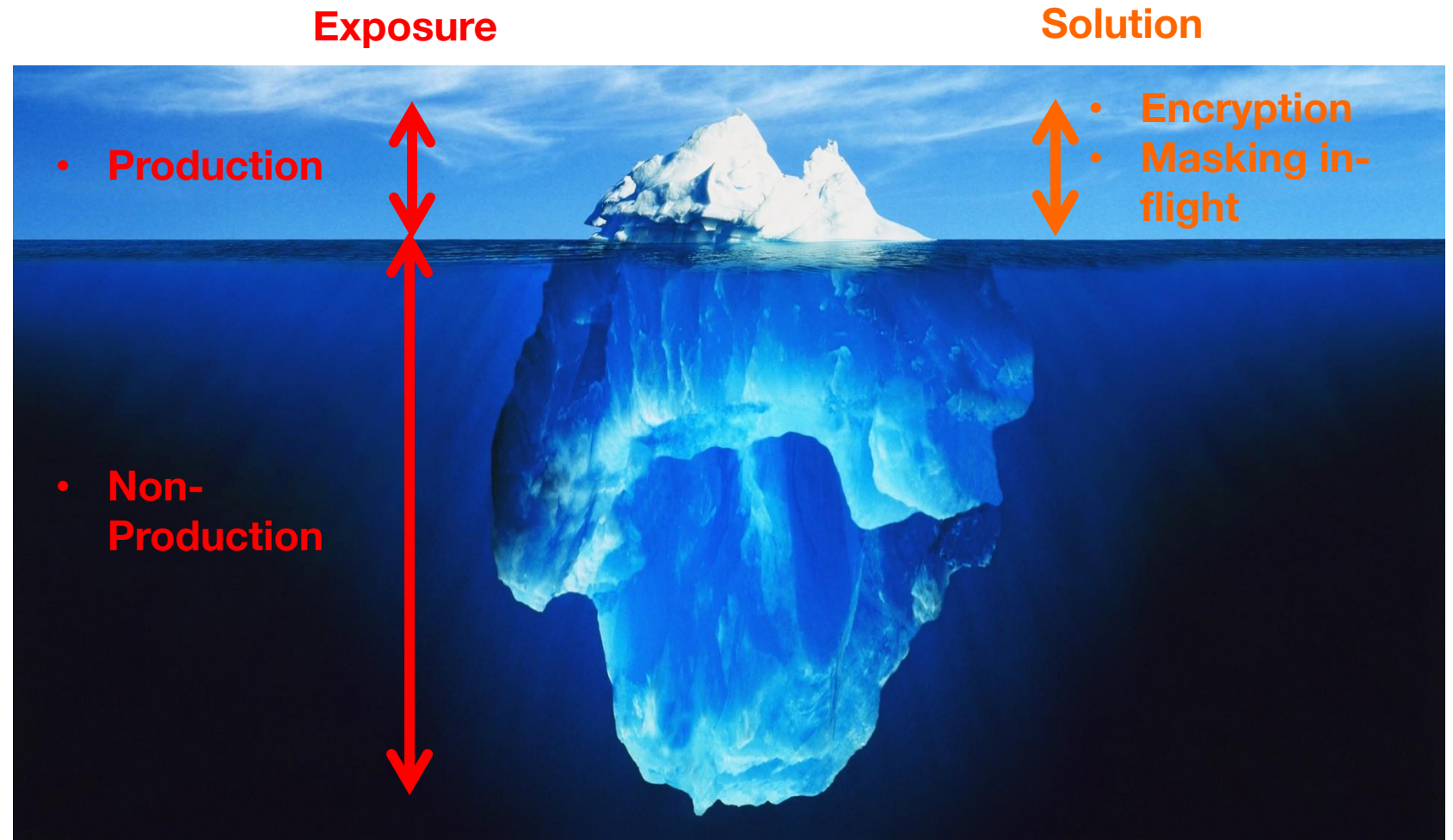
ADVANTAGES

- ▶ Effective for obfuscating data in production systems by not changing data at-rest

- SQL Server Dynamic Data Masking (DDM) is an example

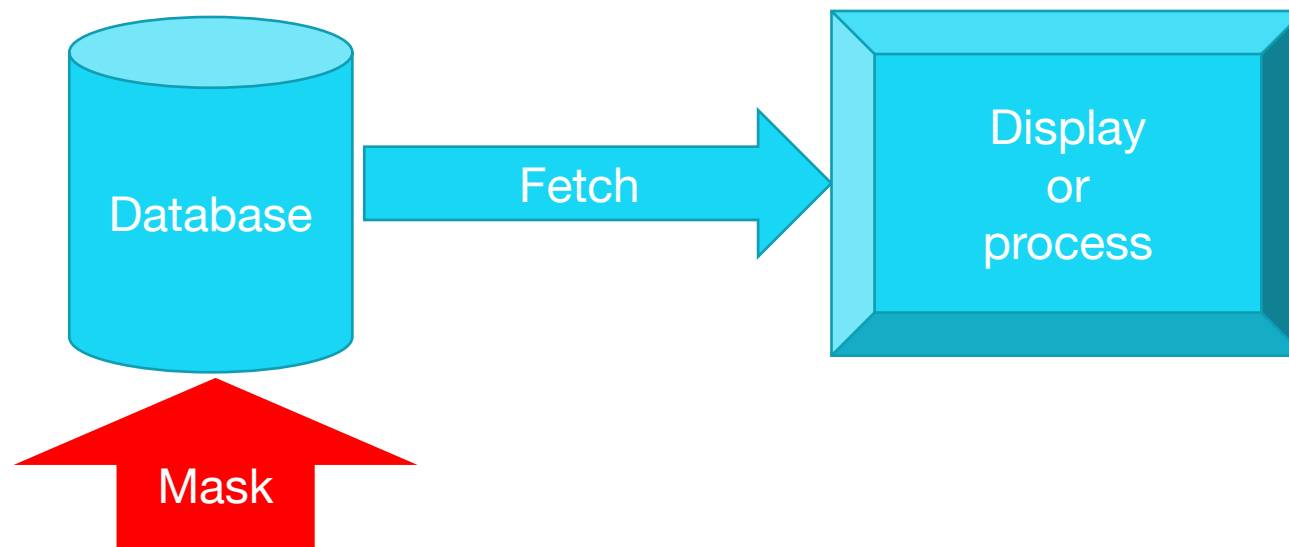
Data masking

- Encryption is the appropriate solution in **production** systems
 - *obfuscation* which is *reversible* upon *authorization*



External and internal threats

- **Masking data at-rest** is the obfuscation of data within the database using SQL statements
- Masking = ***non-reversible*** obfuscation



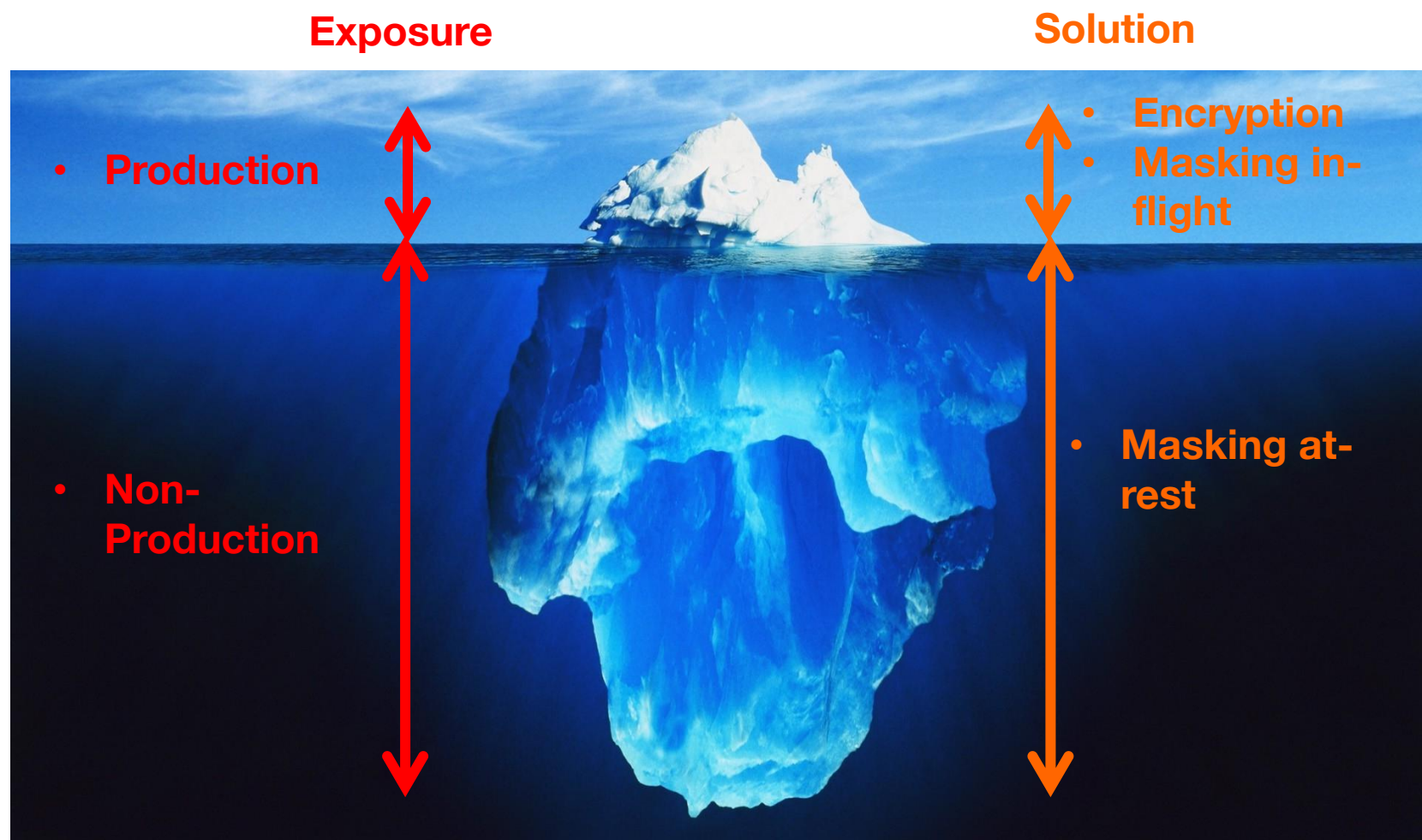
- Delphix, IBM Optim, Informatica data masking are examples

ADVANTAGES

- ▶ Effective for obfuscating data in non-production systems by changing data at-rest
- ▶ Allows provisioning non-production systems outside of secured authorized environments

Data masking

- **Encryption** is the appropriate solution in **production** systems
 - *obfuscation* which is *reversible* upon *authorization*
- **Masking** is the appropriate solution in **non-production** systems
 - *obfuscation* which is *never reversible*



Data masking

- Data masking is the *permanent irreversible* obfuscation of data

Data masking

- Data masking is the *permanent irreversible* obfuscation of data
 - Obfuscation **does not** always mean *scrambling* or *randomizing*
 - Scrambled data looks *awful*, so awful that it is distracting

Data masking

- Data masking is the *permanent irreversible* obfuscation of data
 - Obfuscation **does not** always mean *scrambling* or *randomizing*
 - Scrambled data looks *awful*, so awful that it is distracting
 - Obfuscation **does** involve the use of sophisticated algorithms to irreversibly obfuscate data
 - So that the data looks *useful*, but has no relation to the original values

Data masking

- Data masking is the *permanent irreversible* obfuscation of data
 - Obfuscation **does not** always mean *scrambling* or *randomizing*
 - Scrambled data looks *awful*, so awful that it is distracting
 - Obfuscation **does** involve the use of sophisticated algorithms to irreversibly obfuscate data
 - So that the data looks *useful*, but has no relation to the original values
- Many sensitive data items have embedded encoding
 - Simply scrambling these fields will cause applications to break

Data masking

- Data masking is the *permanent irreversible* obfuscation of data
 - Obfuscation **does not** always mean *scrambling* or *randomizing*
 - Scrambled data looks *awful*, so awful that it is distracting
 - Obfuscation **does** involve the use of sophisticated algorithms to irreversibly obfuscate data
 - So that the data looks *useful*, but has no relation to the original values
- Many sensitive data items have embedded encoding
 - Simply scrambling these fields will cause applications to break
- Many sensitive data items are related groupings
 - Algorithms must be available to mask groups of fields

Data masking

- Data masking is the *permanent irreversible* obfuscation of data
 - Obfuscation **does not** always mean *scrambling* or *randomizing*
 - Scrambled data looks *awful*, so awful that it is distracting
 - Obfuscation **does** involve the use of sophisticated algorithms to irreversibly obfuscate data
 - So that the data looks *useful*, but has no relation to the original values
- Many sensitive data items have embedded encoding
 - Simply scrambling these fields will cause applications to break
- Many sensitive data items are related groupings
 - Algorithms must be available to mask groups of fields
- Many data items can be inferred from other data items
 - Suppose there are only nine of 5000 patients standing taller than 2 meters in the original data
 - Having one or more known items of data can permit the inference of identity

Data masking

- **Secure Lookup**
 - One of eight (8) data transformation frameworks pre-built into the Delphix masking engine
 - Patented proprietary encrypt / hash / modulus lookup algorithm, repeatable yet unbreakable
 - Used to assign a realistic value from a value selected from a pre-defined lookup table
 - The algorithm is irreversible and purposely creates collisions in the output values for added security

Data masking

- Secure Lookup

- One of eight (8) data transformation frameworks pre-built into the Delphix masking engine
 - Patented proprietary encrypt / hash / modulus lookup algorithm, repeatable yet unbreakable
- Used to assign a realistic value from a value selected from a pre-defined lookup table
 - The algorithm is irreversible and purposely creates collisions in the output values for added security

- Example

1. Starting with original column value of “XYZ Holdings”
2. original table has about 1000 distinct data values in the column
 - lookup table can be defined with 500 distinct data values
3. Encrypt original value using AES 256 to “1Gq1159bm7aX2C3bBVMJ3uIg%=”
4. MD5 Hash of the encrypted result = “428618117”
5. $428618117 \bmod 500 = 117$
6. Value within lookup table at entry 117 is “Standard Oil”

Data masking

- Encryption is a solution for production systems
 - Where all users are authorized by the application
- Encryption is not a complete solution for non-production systems
 - Few if any users are authorized at all
- Data masking is the right solution for non-production systems
 - Irreversibly make sensitive data inconsequential from a security perspective, yet realistic
- Providing sets of data quickly to developers and testers is necessary
 - But exposing sensitive data to unauthorized consumers will come with fines and other punitive measures
- Data virtualization solutions must be designed for both agility and security together
 - Must be expandable for localization and customization

Summary

Data is the constraint for testing and development

Summary

Data is the constraint for testing and development

- **Data virtualization** is a new use of proven technologies combined
 - 100% of all IT shops doing software development need data virtualization

Summary

Data is the constraint for testing and development

- **Data virtualization** is a new use of proven technologies combined
 - 100% of all IT shops doing software development need data virtualization
- **Data masking** reduces the surface area of risk

Summary

Data is the constraint for testing and development

- **Data virtualization** is a new use of proven technologies combined
 - 100% of all IT shops doing software development need data virtualization
- **Data masking** reduces the surface area of risk
- Together **data virtualization** and **data masking** accelerates testing and development securely
 - On-premise
 - In the IaaS cloud

Q & A



Tim.Gorman@Delphix.com



[@TimGormanTech](https://twitter.com/TimGormanTech)



<https://www.linkedin.com/in/TimGorman/>