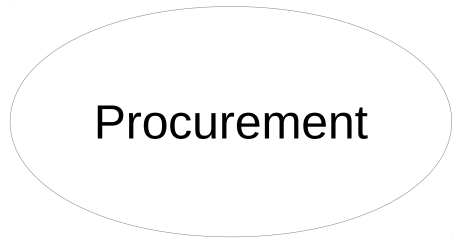
Network Bound Encryption for Data-at-Rest Protection

Nathaniel McCallum,
Principal Software Engineer
Red Hat, Inc.

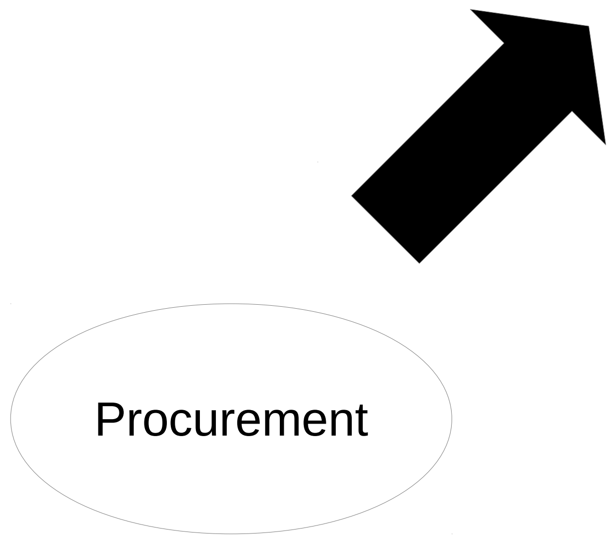
redhat.

Disk Life Cycle

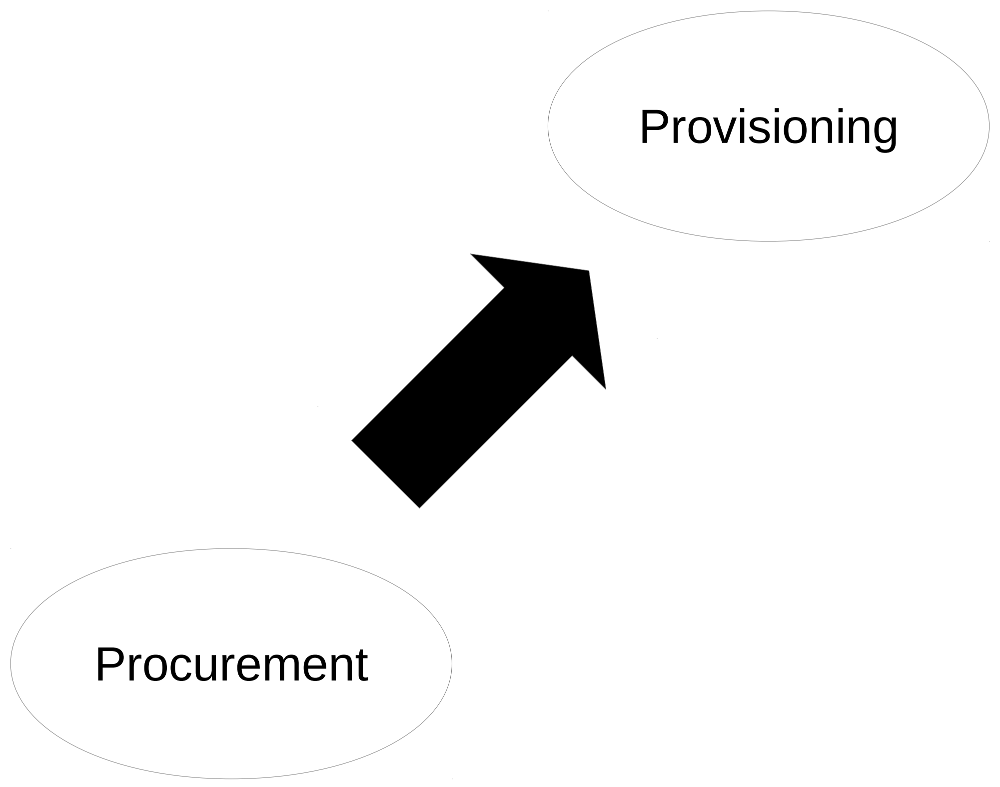
Disk Life Cycle



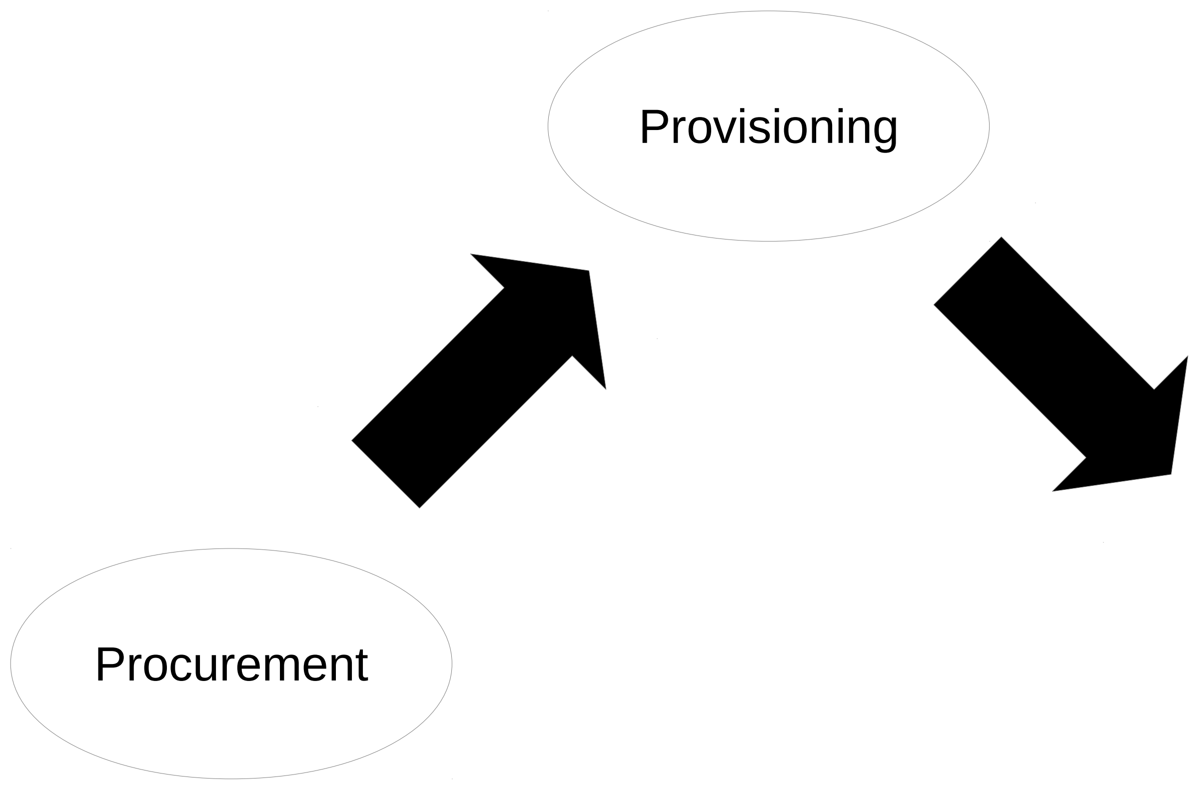
Disk Life Cycle



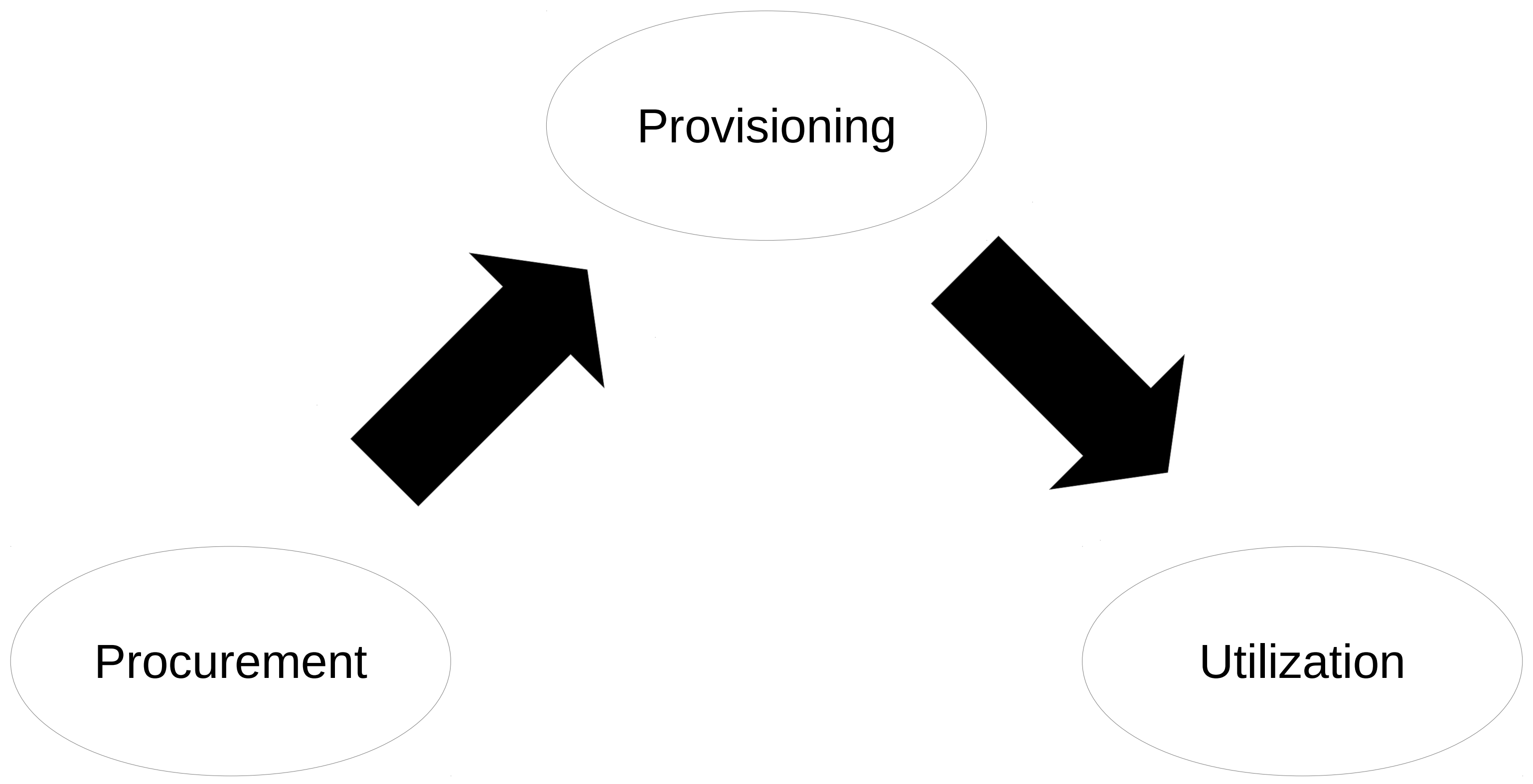
Disk Life Cycle



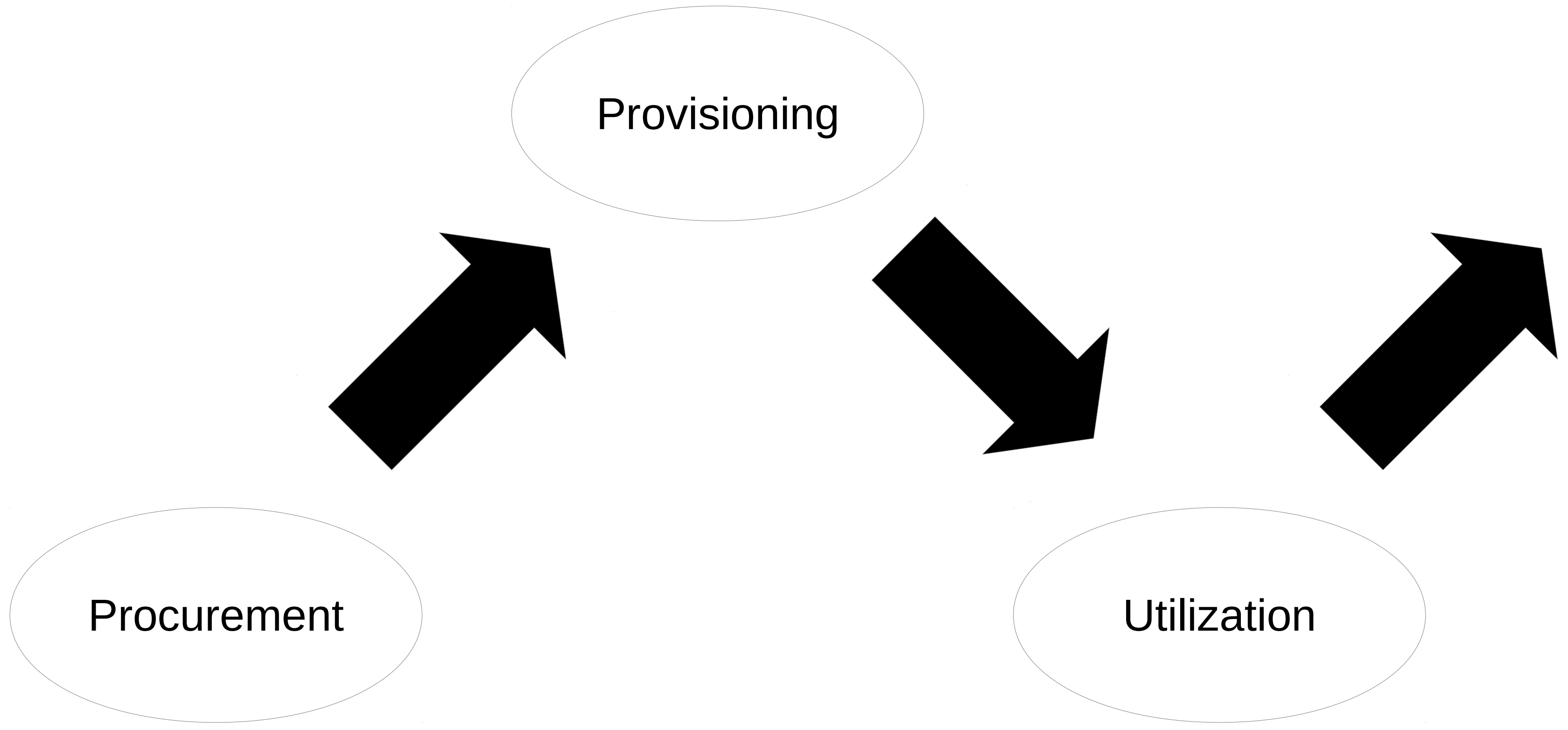
Disk Life Cycle



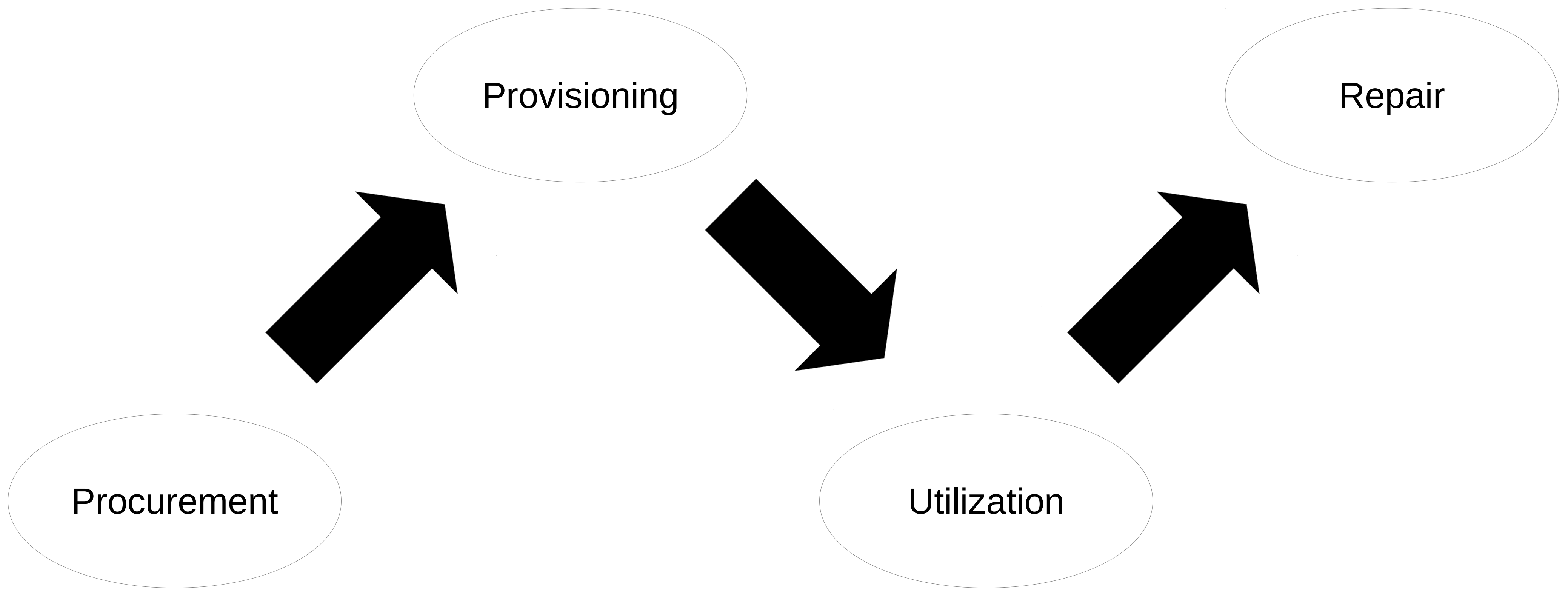
Disk Life Cycle



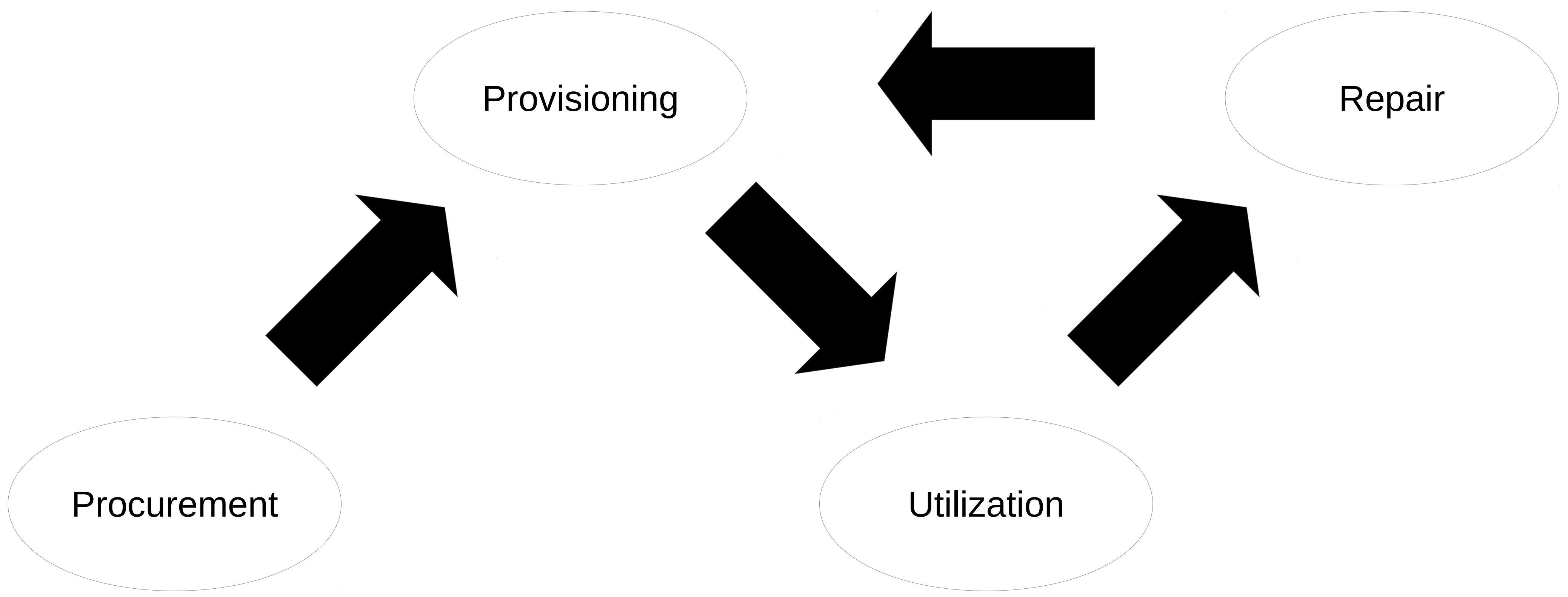
Disk Life Cycle



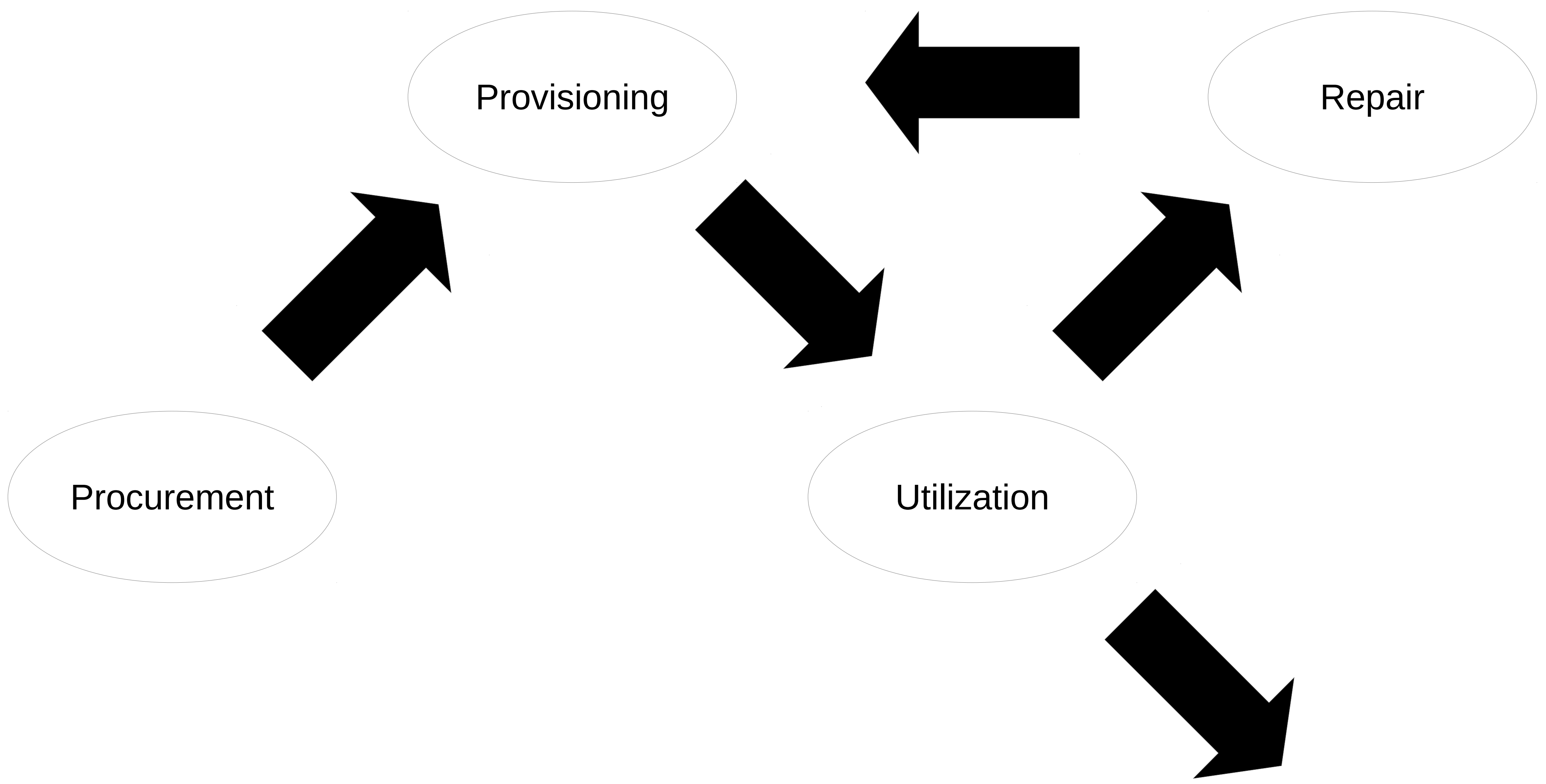
Disk Life Cycle



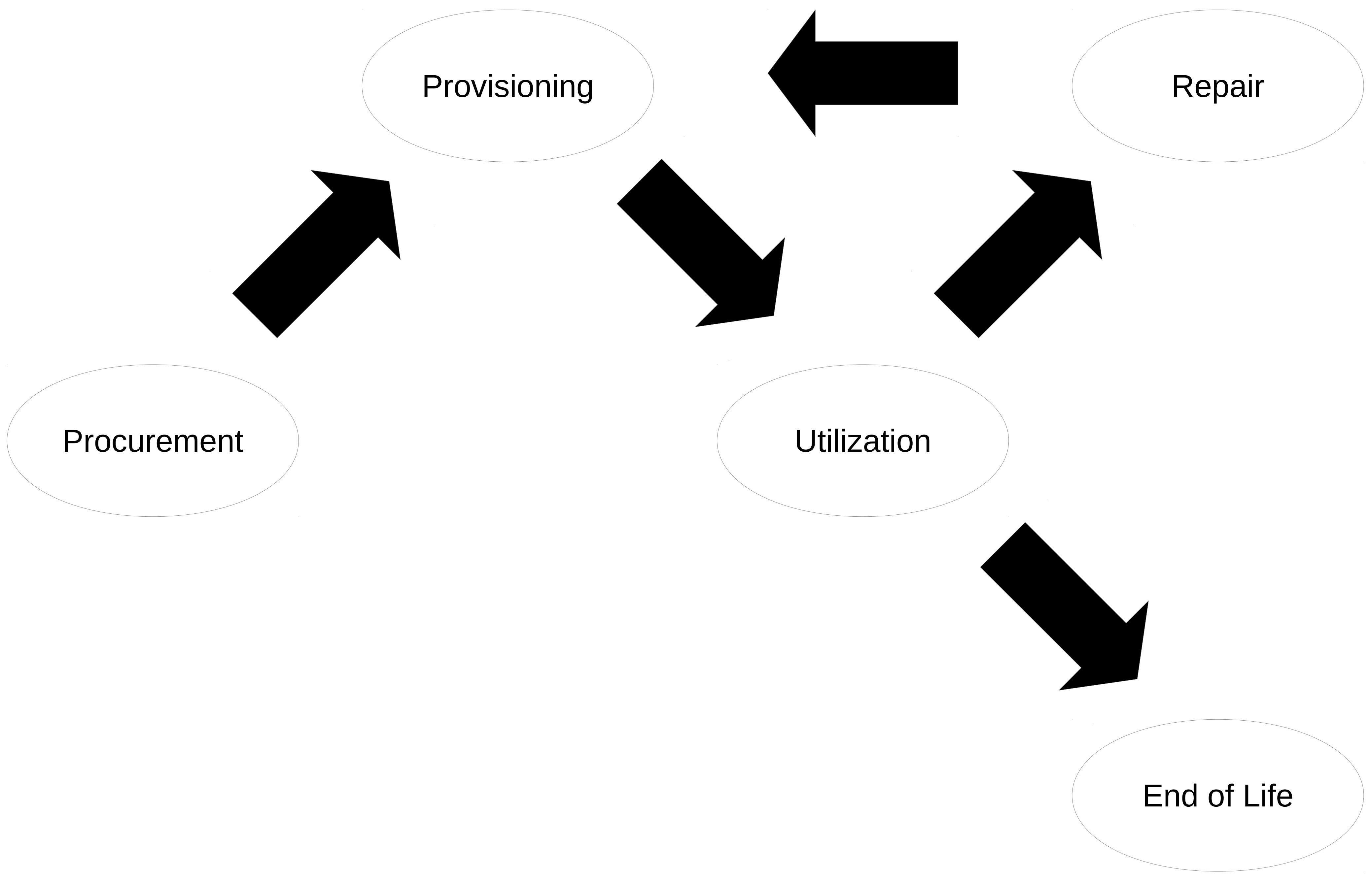
Disk Life Cycle



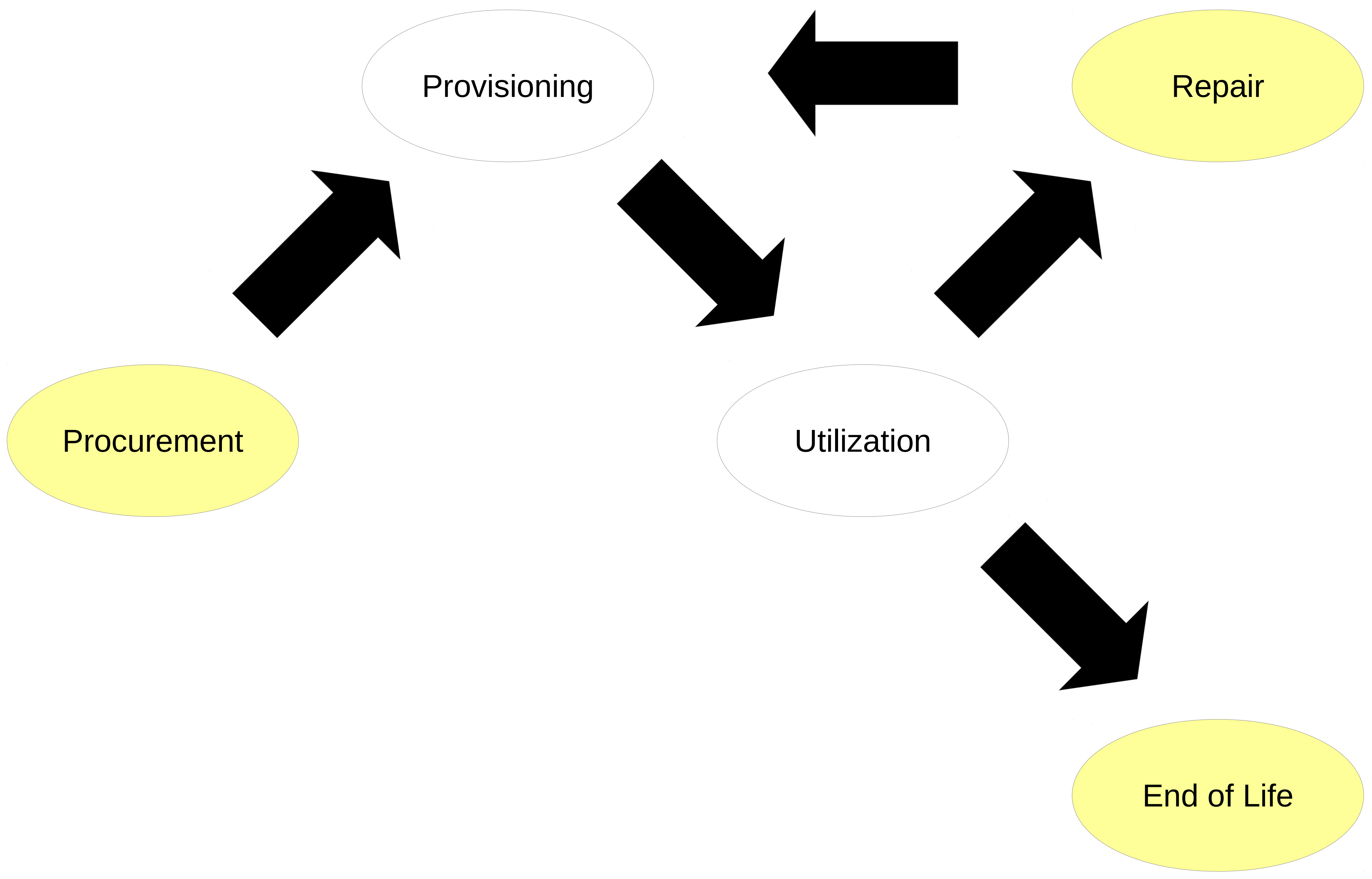
Disk Life Cycle



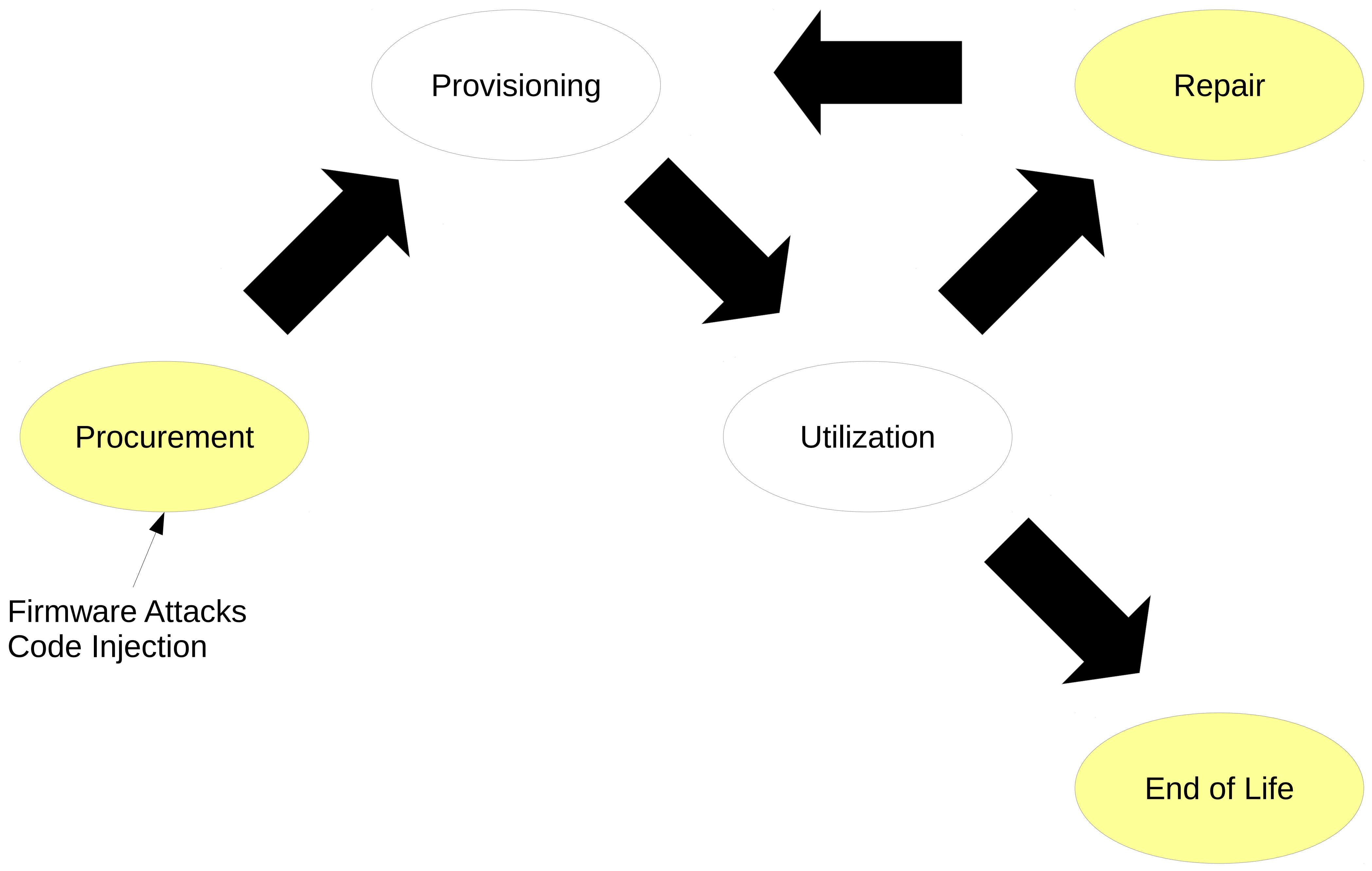
Disk Life Cycle



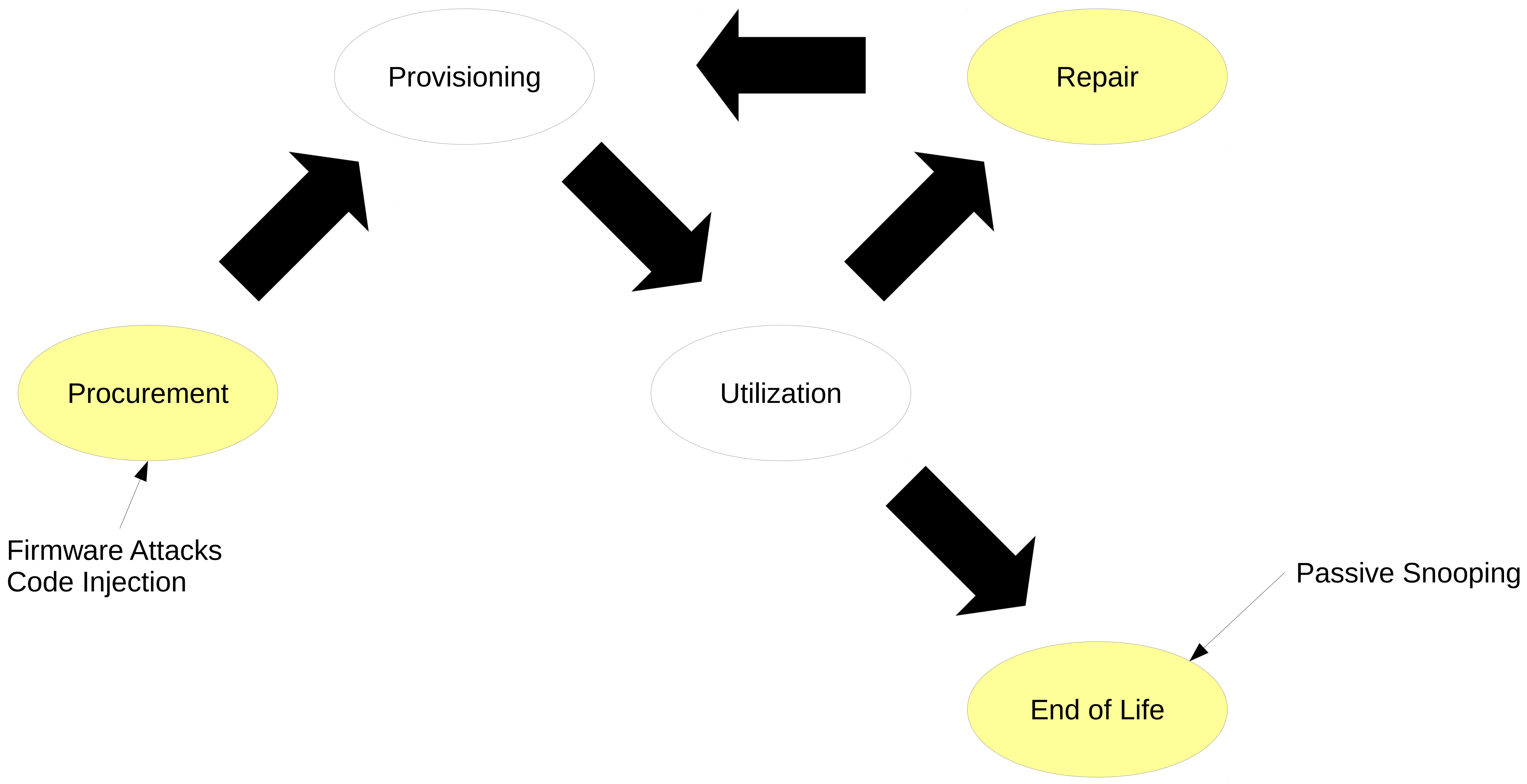
Disk Life Cycle (Third Parties)



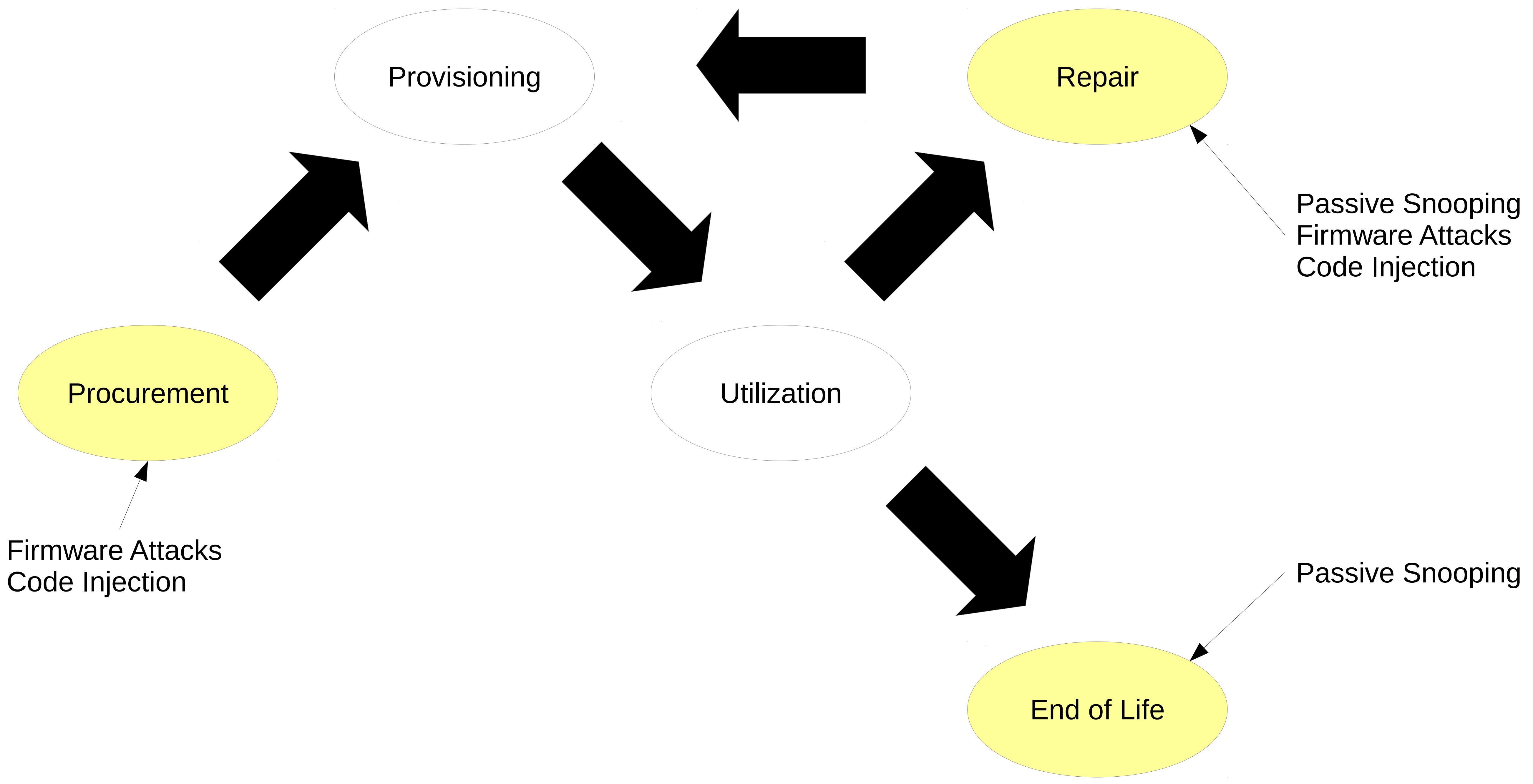
Disk Life Cycle (Third Parties; Threat Models)



Disk Life Cycle (Third Parties; Threat Models)



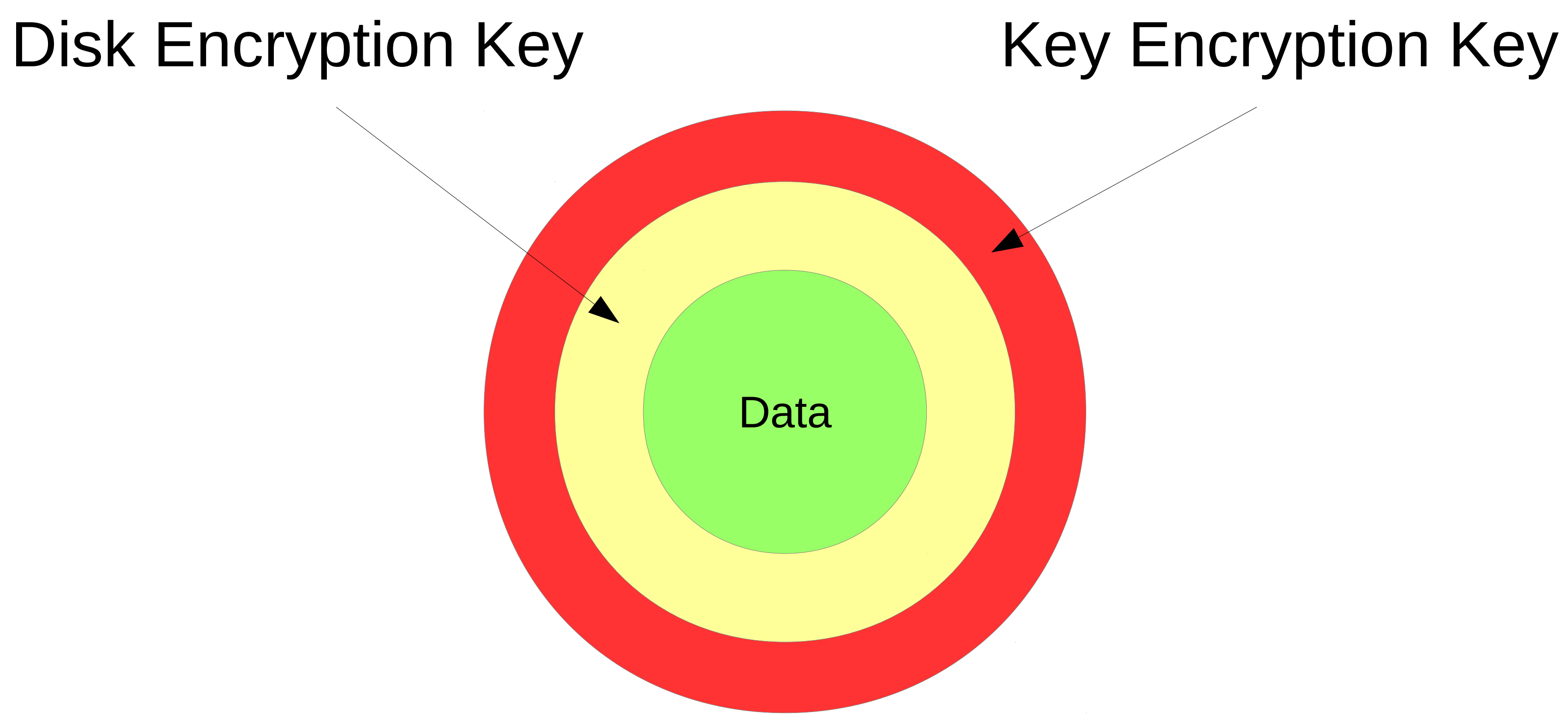
Disk Life Cycle (Third Parties; Threat Models)



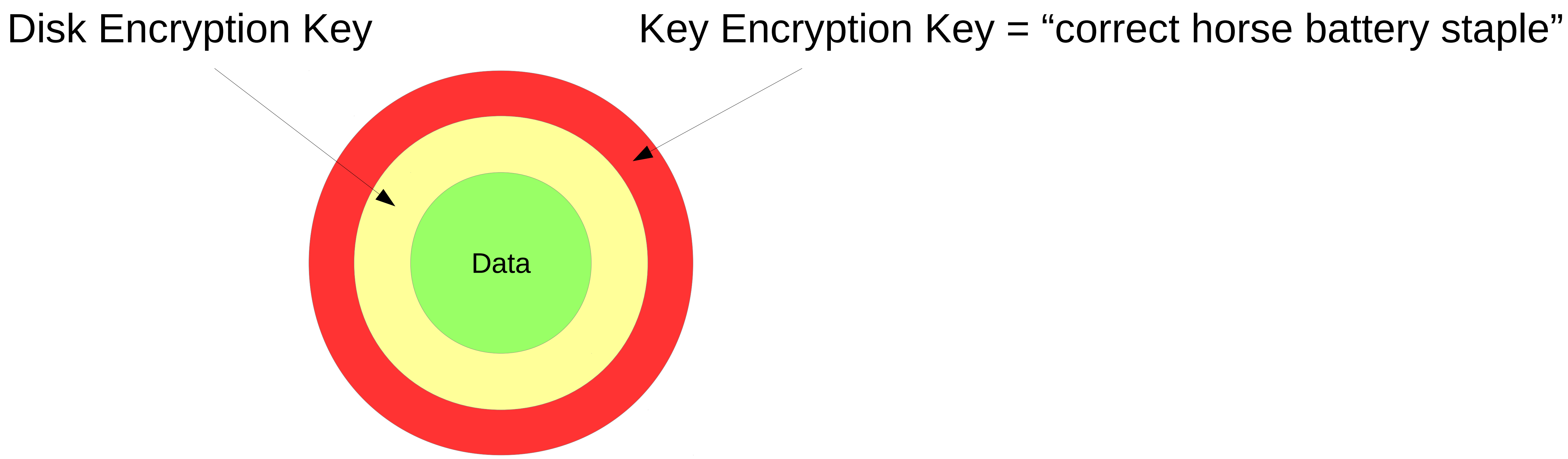
Security Threats and Mitigation

- Provisioning:
  - Firmware Attacks – Flash Firmware
  - Code Injection – Format (Restore)
- Encryption:
  - Passive Snooping
  - Problem: Key Management?

Key Usage Overview: Symmetric Encryption



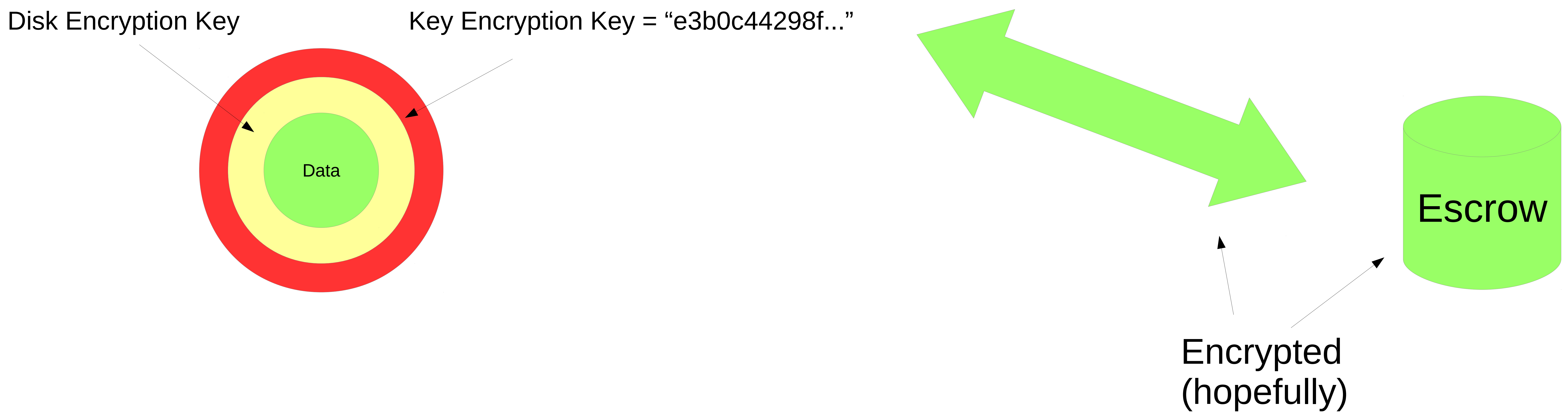
Common Technique #1: Shared KEKs



Common Technique #1: Shared KEKs

Pros	Cons	
Easy (A.K.A. procrastination)	Manual Unlocking	Manual Rotation
Early Boot	Granularity vs. Scalability	No Access Control
	Vulnerable to Ex-Employees	Vulnerable to Social Hacking

Common Technique #2: KEK Escrow / Retrieval

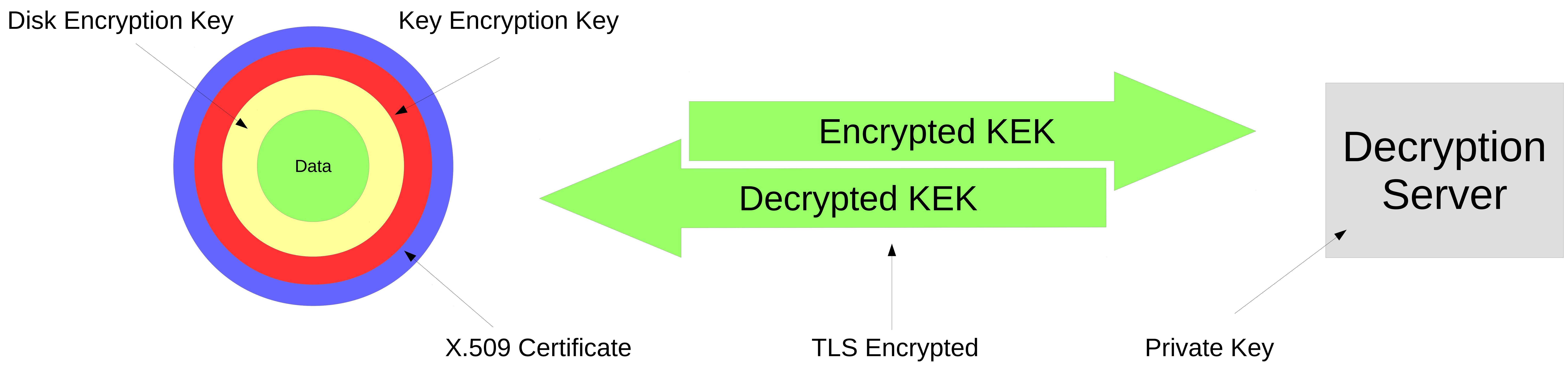


Common Technique #2: KEK Escrow / Retrieval

Pros	Cons	
Key Separation	No Offline Escrow	Manual Rotation (or CRL)
Automatic Unlocking	Key First, Disk Later	Requires Authentication
Access Control	“Key in Tunnel” Design	Stateful Server
Auditing	(Usually) No Early Boot	

Can we use asymmetric crypto to improve the situation?

New Technique #1: X.509



New Technique #1: X.509

Pros		Cons
Key Separation	No Key First, Disk Later	“Key in Tunnel” Design
Automatic Unlocking	No Authentication Required	Manual Rotation (or CRL)
Offline Escrow		Limited Access Control
Early Boot		Limited Auditing
Stateless Server		Difficult to Configure (X.509)

New Technique #1: X.509

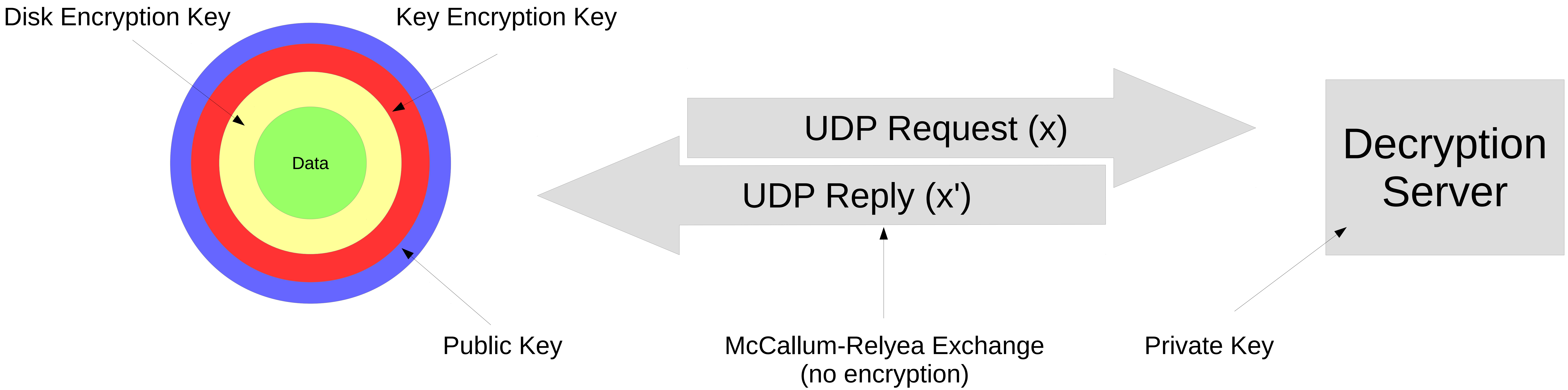
- Most major drawbacks relate to the use of X.509
- Can we shrink implementation requirements for embedded use?
- Can we hide key contents from the Decryption Server?
- Can we avoid TLS?

New Technique #1: X.509

- Most major drawbacks relate to the use of X.509
- Can we shrink implementation requirements for embedded use?
- Can we hide key contents from the Decryption Server?
- Can we avoid TLS?

- Yes!

New Technique #2: McCallum-Relyea Exchange



Elgamal Encryption

Group Parameters: p, g

Encryption

Step	Client	Server
1		$B \in_R [1, p-1]$
2		$b = g^B$
3	$\longleftarrow b$	
4	$K \in_R \mathbb{Z}_p$	
5	$A \in_R [1, p-1]$	
6	$a = g^A,\ k = Kb^A$	

Decryption

Step	Client	Server
1	$a, k \longrightarrow$	
2		$K = k \div a^B$

McCallum Opaque Decryption

Group Parameters: p, g

Encryption

Step	Client	Server
1		$B \in_R [1, p-1]$
2		$b = g^B$
3	$\longleftarrow b$	
4	$K \in_R \mathbb{Z}_p$	
5	$A \in_R [1, p-1]$	
6	$a = g^A,\ k = Kb^A$	

Decryption

Step	Client	Server
1	$X \in_R [1, p-1]$	
2	$k' = kg^X$	
3		$a, k' \longrightarrow$
4		$k'' = k' \div a^B$
5		$\longleftarrow k''$
6	$K = k'' \div g^X$	

McCallum-Relyea Exchange

Group Parameters: p, g

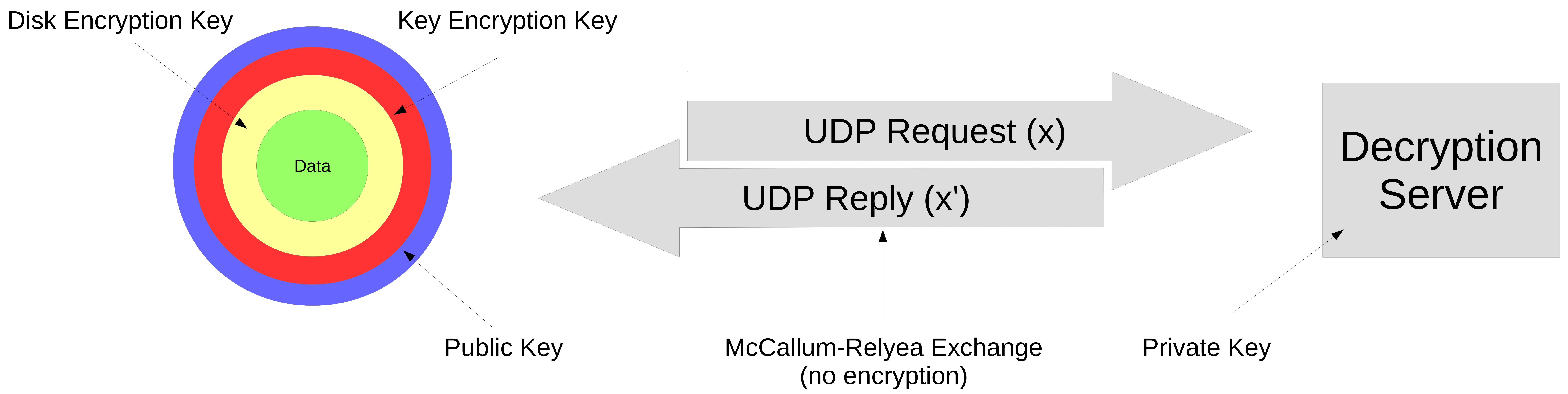
Encryption

Step	Client	Server
1		$B \in_R [1, p-1]$
2		$b = g^B$
3	$\longleftarrow b$	
4	$K \in_R \mathbb{Z}_p$	
5	$A \in_R [1, p-1]$	
6	$a = g^A,\ k = Kb^A$	

Decryption

Step	Client	Server
1	$X \in_R [1, p-1]$	
2	$x = ag^X$	
3	$x \longrightarrow$	
4		$x' = x^B$
5	$\longleftarrow x'$	
6	$K = k \div (x' \div b^X)$	

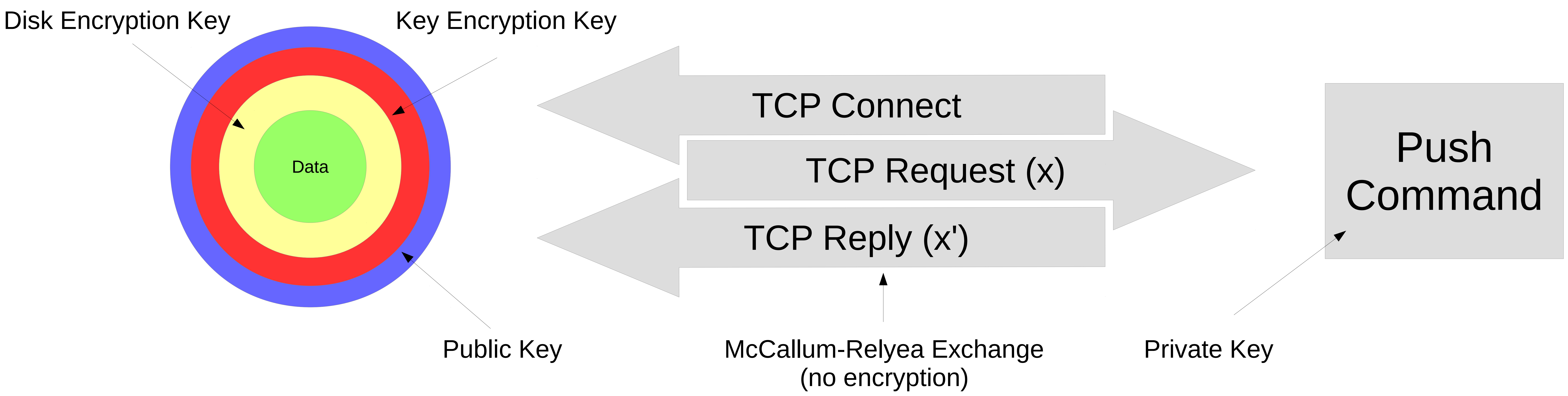
New Technique #2: McCallum-Relyea Exchange



New Technique #2: McCallum-Relyea Exchange

Pros		Cons
Key Separation	No Key First, Disk Later	Limited Access Control
Automatic Unlocking	No Authentication Required	Limited Auditing
Offline Escrow	“Key in Tunnel” Design	
Early Boot	Automatic Rotation	
Stateless Server		

New Technique #3: Push McCallum-Relyea Exchange



New Technique #3: Push McCallum-Relyea Exchange

Pros		Cons
Key Separation	No Key First, Disk Later	
Automatic Unlocking	No Authentication Required	
Offline Escrow	“Key in Tunnel” Design	
Early Boot	Automatic Rotation	
Stateless Server	Access Control, Auditing	

Upstream Project: Deo

- https://github.com/npmccallum/deo
- Δεο: to bind
- Project Status: Unstable
  - Technique #1 implemented (X.509; deprecated – don't use)
  - Techniques #2 and #3 in development
  - Early boot (LUKS) implemented
  - Support for ext4 crypto in planning
- Contributions welcome!

redhat®