



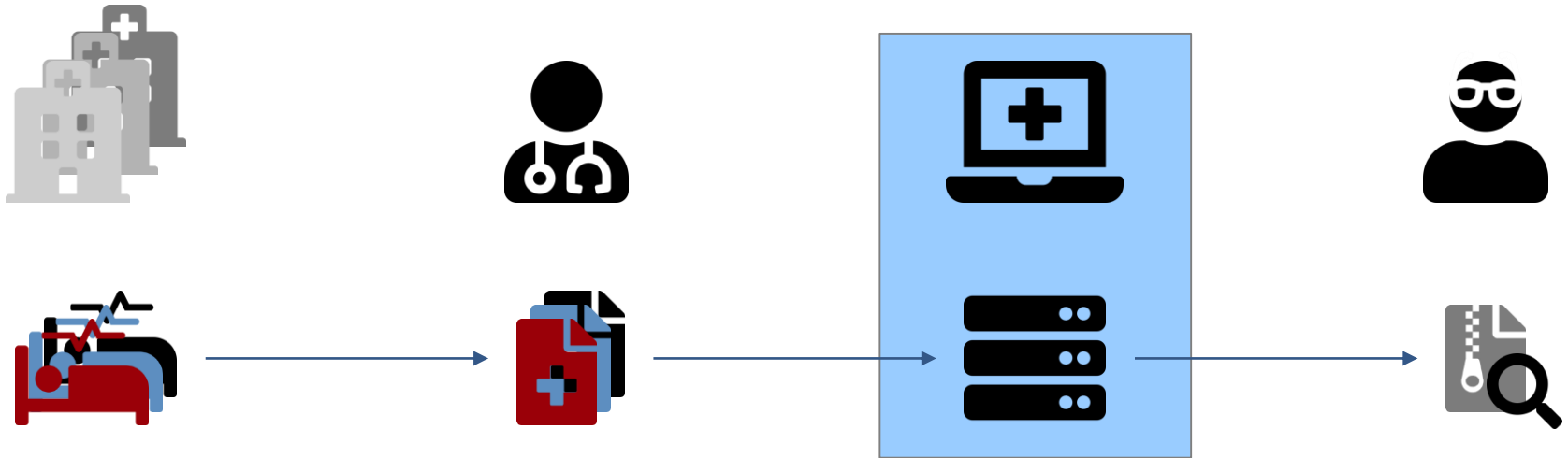
Datenschutzgerechte und mehrseitig sichere IT-Plattformen für die medizinische Forschung

Tom Petersen
Sicherheit in Verteilten Systemen (SVS)
Universität Hamburg

Einführung: Medizinische Forschung



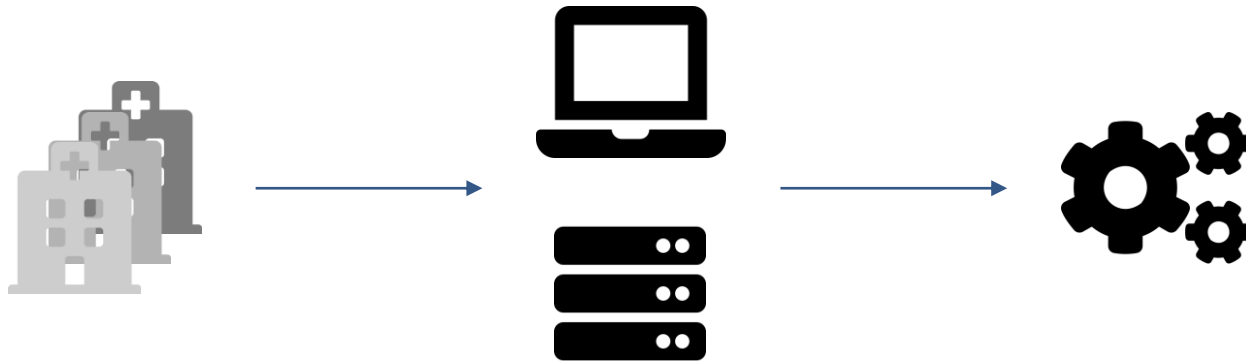
Einführung: Medizinische Forschung



Plattform

- Mehrseitig sicher
- Datenschutzgerecht
- Verschiedene Architekturen
- Technische Maßnahmen

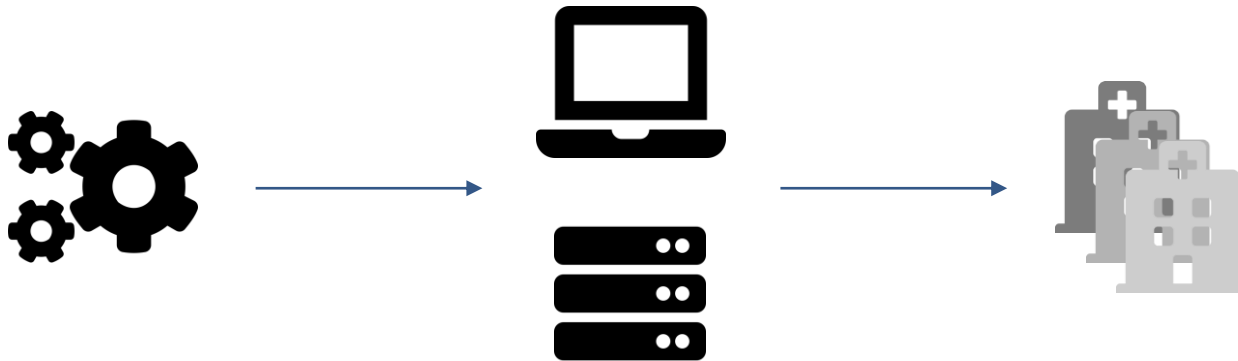
Neue Ausrichtung



Working Title:

Datenschutzgerechte und mehrseitig sichere IT-Plattformen für die medizinische Forschung

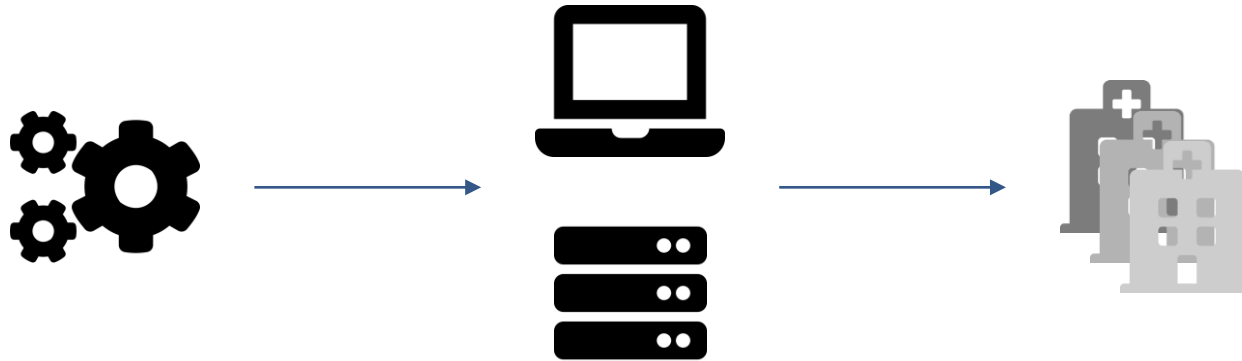
Neue Ausrichtung



Working Title:

Sichere Bausteine für kritische Anwendungen

Neue Ausrichtung



Working Title:

Sichere Bausteine für kritische Anwendungen

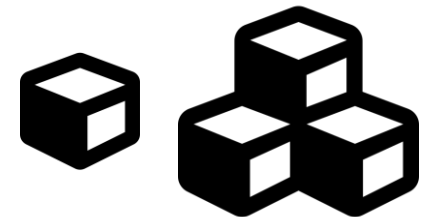
1. Auswahl von Bausteinen
2. Design nutzbarer Sicherheits-APIs
3. Verschiedene methodische Arbeiten im Bereich bestimmter Bausteine
4. Umsetzung
5. Einsatz im Kontext von Plattformen für die medizinische Forschung

Auswahl von Bausteinen

- Bausteine: Für sichere Software relevante Primitive, die von unterliegenden kryptographischen Primitiven abstrahieren

- Beispiele:

- Erstellung von sicheren Zufallswerten
- Suche auf verschlüsselten Daten
- Sichere Speicherung von Passwörtern:



```
password_hash(pw)
```

statt

```
Argon2 („ID“, 4, salt, pw)
```

- Wie können relevante Bausteine für sichere und datenschutzgerechte Entwicklung ausgewählt werden? Ableitung aus:
 - PbD-/SbD-Prinzipien oder -Patterns,
 - rechtlichen Anforderung, bspw. der DSGVO,
 - häufigen Schwachstellen, bspw. der unsicheren Speicherung von Passwörtern, ...
 - ...

Design nutzbarer Sicherheits-APIs

- Wie können APIs derart gestaltet werden, dass fehlerbehaftete Entwicklung erschwert wird?
- Eigenschaften :
 - Einfach nutzbar
 - Einfach zu erlernen
 - Sichere Standardwerte
 - Schwierig, falsch zu benutzen
 - ...



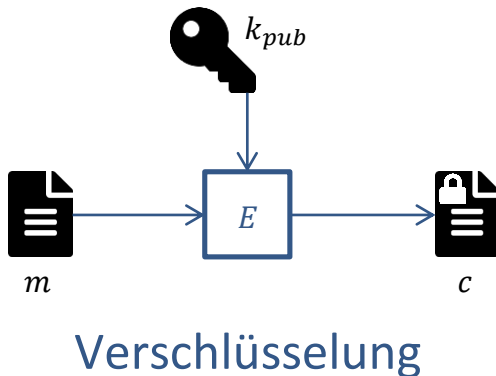
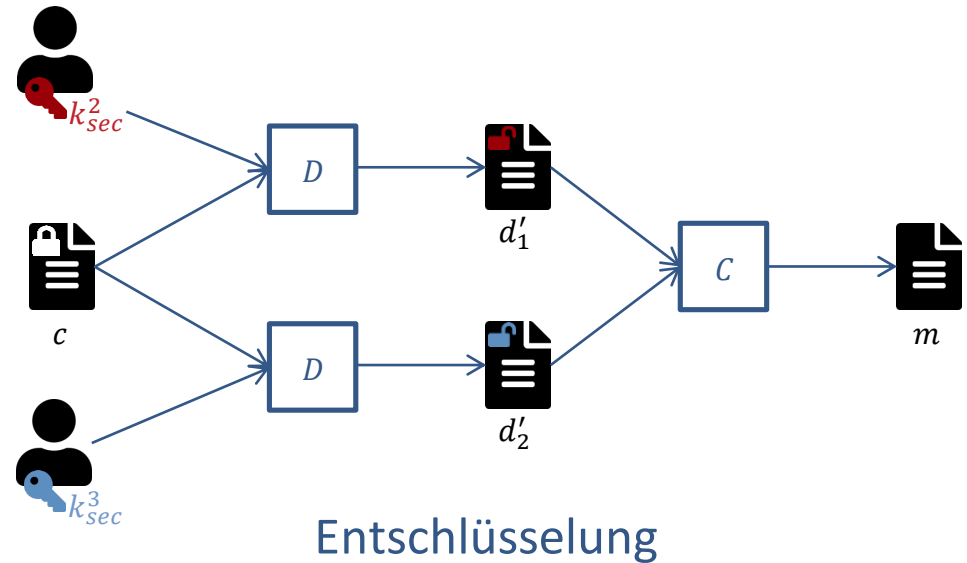
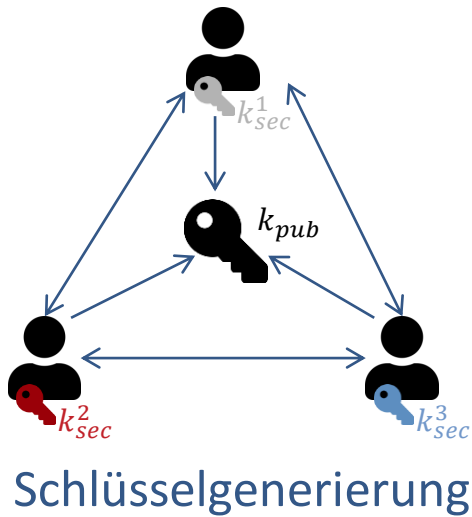
```
SecretKeySpec keySpec = newSecretKeySpec(raw, "AES");  
Cipher cipher = Cipher.getInstance("AES");  
cipher.init(Cipher.ENCRYPT_MODE, keySpec);  
byte[] encrypted = cipher.doFinal(clear);
```

Unsichere Standardwerte in der Java Cryptography Architecture (JCA).

Methodische Arbeiten im Bereich bestimmter Bausteine

1. **Mehraugenprinzip** durch kryptographische Schwellwertschemata
2. **Need-to-know-Prinzip** durch Searchable-Encryption-Schemata
3. **Anonyme Datenveröffentlichung** unter geeigneten Metriken

Mehraugenprinzip durch kryptographischen Schwellwertschemata



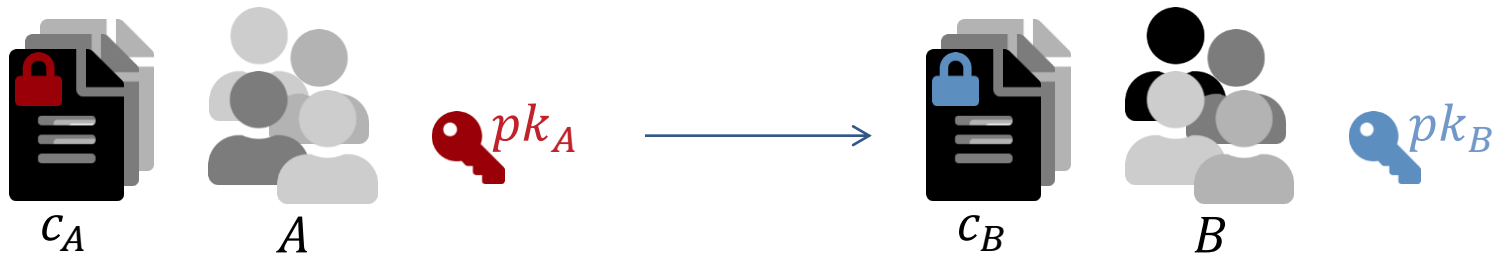
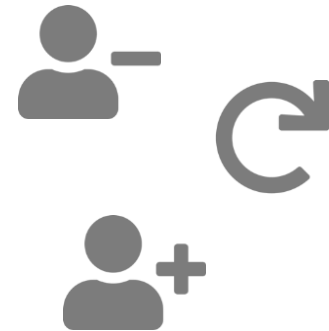
Nutzung des **Mehraugenprinzips** für:

- Pseudonymaufdeckung
- Sensible Logdaten
- ...

Schlüsselmanagement für Schwellwertschemata

■ Situationen

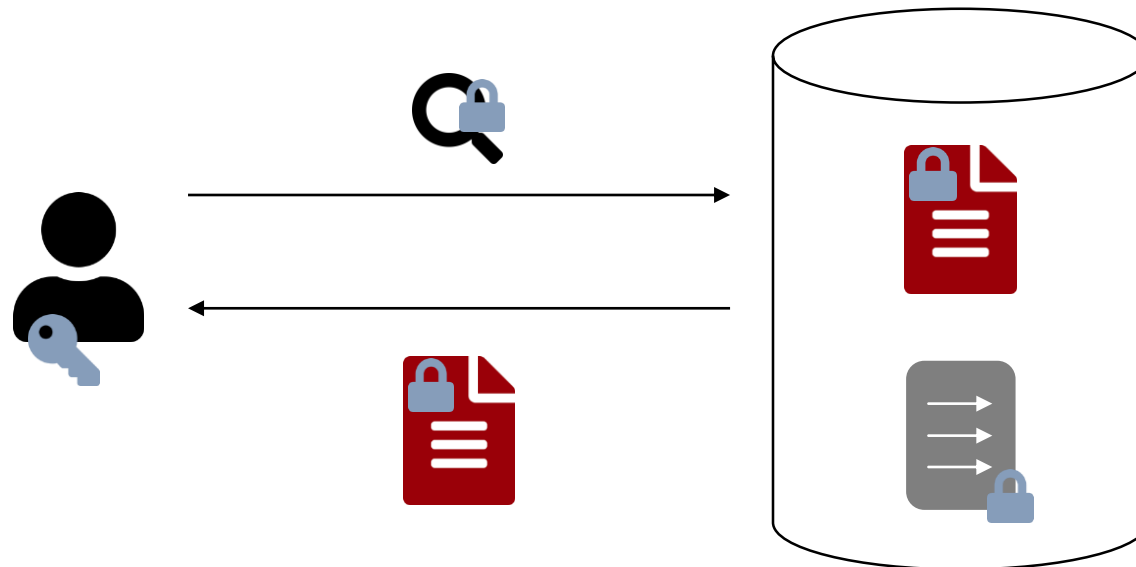
- Hinzufügen oder Entfernen eines Teilnehmers
- Ändern des Schwellwertparameters
- Verlorene oder kompromittierte Schlüsselteile
- Proaktive Schlüsselrotation



■ Angestrebte Eigenschaften

- Keine Aufdeckung existierender Schlüsseltexte
- Keine Offenlegung der privaten Schlüssel oder Schlüsselteile
- Keine Entschlüsselung durch Zugriffsstruktur A nach dem Update
- Keine notwendige Kooperation aller Beteiligten für alle Schlüsseltexte

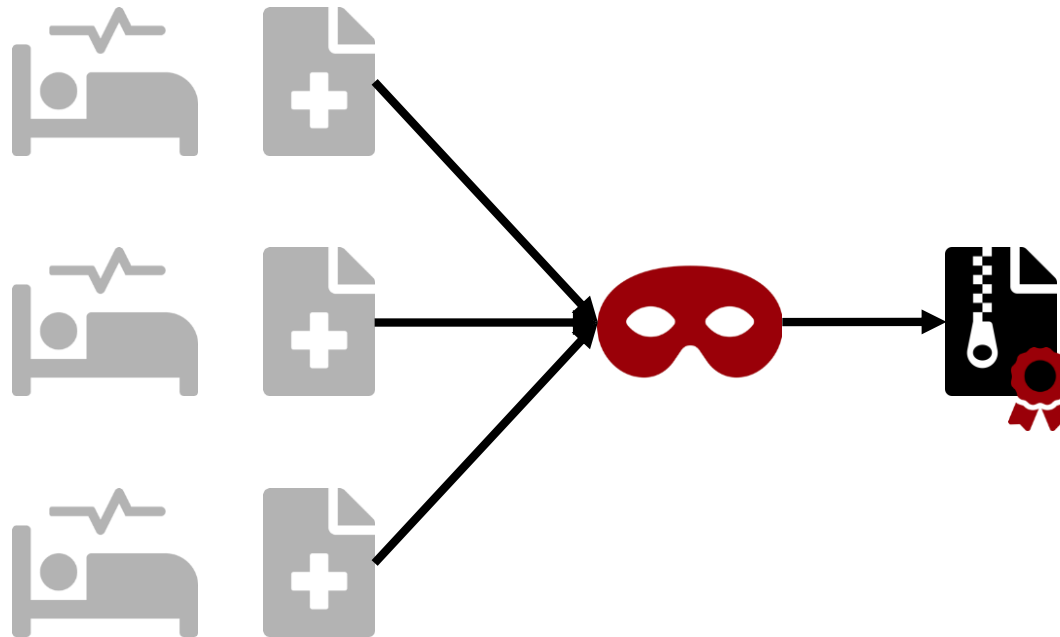
Need-to-know-Prinzip durch Searchable-Encryption-Schemata



■ Eigenschaften

- Arten von Suchanfragen
- Dynamische Indizes
- Rückschlüsse auf unterliegende Daten?
- Performancegewinn?

Datenveröffentlichung unter geeigneten Anonymitätsmetriken



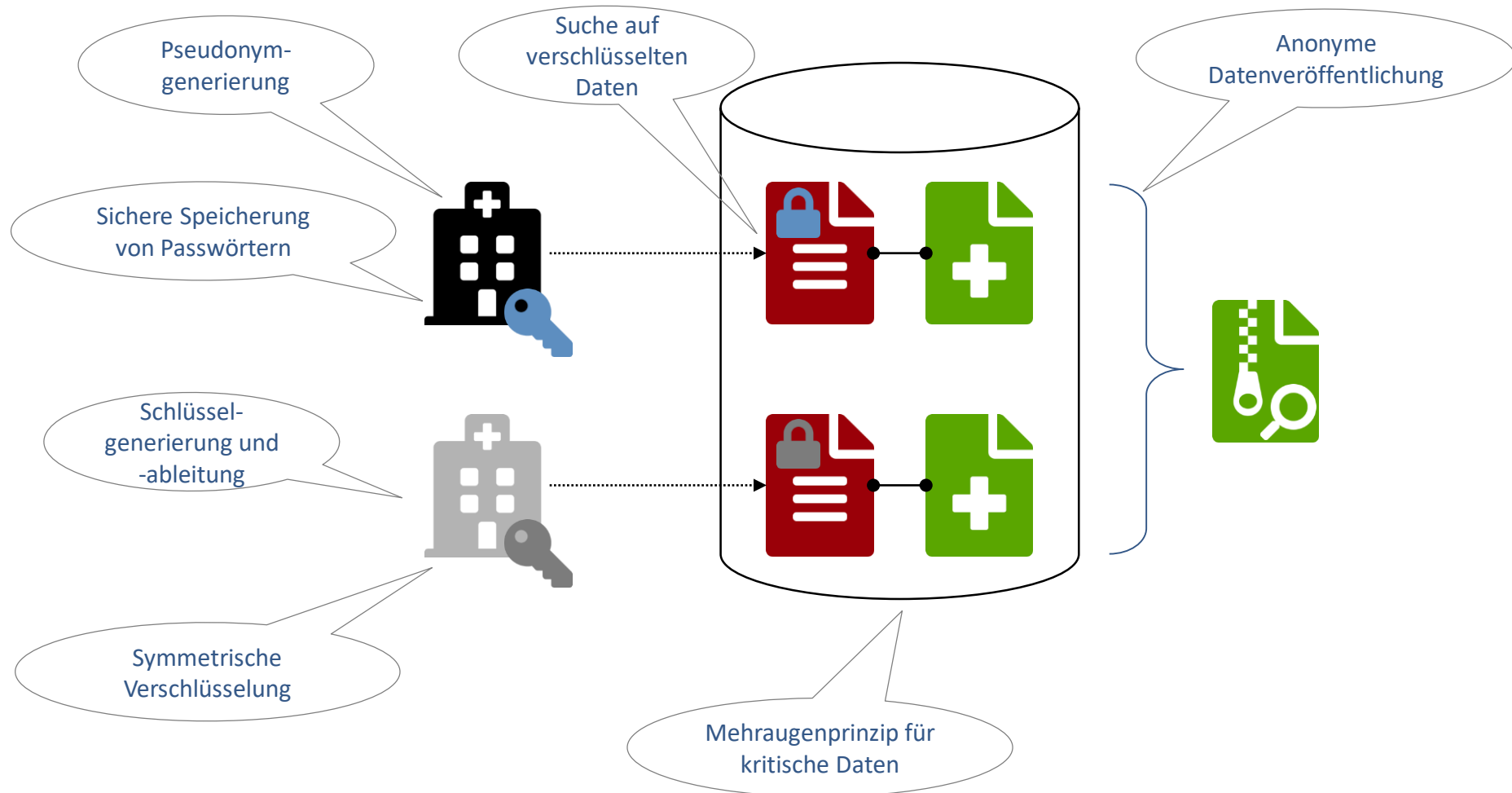
- Eignung verschiedener Metriken
 - k-Anonymität, l-Diversität, t-Closeness, ...
 - Differential Privacy
- Dezentrale, heuristische Ansätze zur Datenanonymisierung

Umsetzung



- Welche Umgebung(-en)?
- Welche Programmiersprache(-n)?
- An geeigneten Stellen Nutzung existierender und überprüfter Krypto-Bibliotheken wie Sodium

Einsatz im Kontext medizinischer Forschungsplattformen



Vielen Dank!

Kommentare oder Fragen?