Bachelorarbeit – Defense
BA21_gueu_09

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

KEY MANAGEMENT AND ENCRYPTION FOR LIVE VIDEOSTREAMS

Nicolas Da Mutten
Daniela Egli
Andreas Meier

Index

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

- Introduction
- Background & Theory
- Methodology & Results
- Discussion
- Outlook
- Demo
- Questions

Introduction

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

MOTIVATION AND CURRENT STATE

- Increased usage of Streaming Services: video encryption increasingly important
- TV in HD: requires encryption by stations
- Concrete use case: Init7

GOALS

- Architectural draft for modular Key Management System
- Document thoughts and decisions tracably
- Develop a prototype

Introduction

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

LITERATURE RESEARCH

- Very little specific literature
- Some sources provided contrary information
- Fairplay specification not entirely conclusive

Index

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

- Introduction
- **Background & Theory**
- Methodology & Results
- Discussion
- Outlook
- Demo
- Questions

Background & Theory

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

STREAMING PROTOCOLS

- Multicast
- Apple HTTP Live Streaming (HLS)
- Dynamic Adaptive Streaming over HTTP (DASH)
- Microsoft Smooth Streaming (MSS)
- Adobe HTTP Dynamic Streaming (HDS)

DRM PROTOCOLS

- Apple FairPlay Streaming
- Google Widevine
- Microsoft PlayReady

Background & Theory

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

COMPATIBILITY

Geräte-kategorie	Player	HLS	DASH	HLS + FairPlay	HLS + Widevine	DASH + Widevine	DASH + PlayReady	MSS + PlayReady
PCs/ Browsers	Chrome	Ja	Ja	Nein	Ja	Ja	Nein	Nein
	Firefox	Ja	Ja	Nein	Ja	Ja	Nein	Nein
	IE/Edge	Ja	Ja	Nein	Nein	Nein	Ja	Ja
	Safari	Ja	Nein	Ja	Nein	Nein	Nein	Nein
Handys	Android	Ja	Ja	Nein	Nein	Ja	Ja	Ja
	iOS	Ja	Nein	Ja	Nein	Nein	Nein	Nein
Set-top Boxen	Chrome-cast	Ja	Ja	Nein	Nein	Ja	Ja	Ja
	Android TV	Ja	Ja	Nein	Nein	Ja	Ja	Ja
	Roku	Ja	Ja	Nein	Nein	Ja	Ja	Ja
	Apple TV	Ja	Nein	Ja	Nein	Nein	Nein	Nein
	Amazon Fire TV	Ja	Ja	Nein	Nein	Ja	Ja	Ja
Smart TVs	Samsung/Tenzen	Ja	Ja	Nein	Nein	Ja	Ja	Ja
	LG webOS	Ja	Ja	Nein	Ja	Nein	Nein	Nein
	SmartTV Alliance	Ja	Ja	Nein	Nein	Nein	Ja	Ja
	Android TV	Ja	Ja	Nein	Nein	Ja	Ja	Ja

Table adapted from Yuriy Reznik, Jordi Cenzano und Bo Zhang. „Transitioning Broadcast to Cloud". In: *Applied Sciences* 11.2 (Jan. 2021). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, S. 503. DOI: 10.3390/app11020503. URL: https://www.mdpi.com/2076-3417/11/2/503 (besucht am 24. 02. 2021)

Index

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

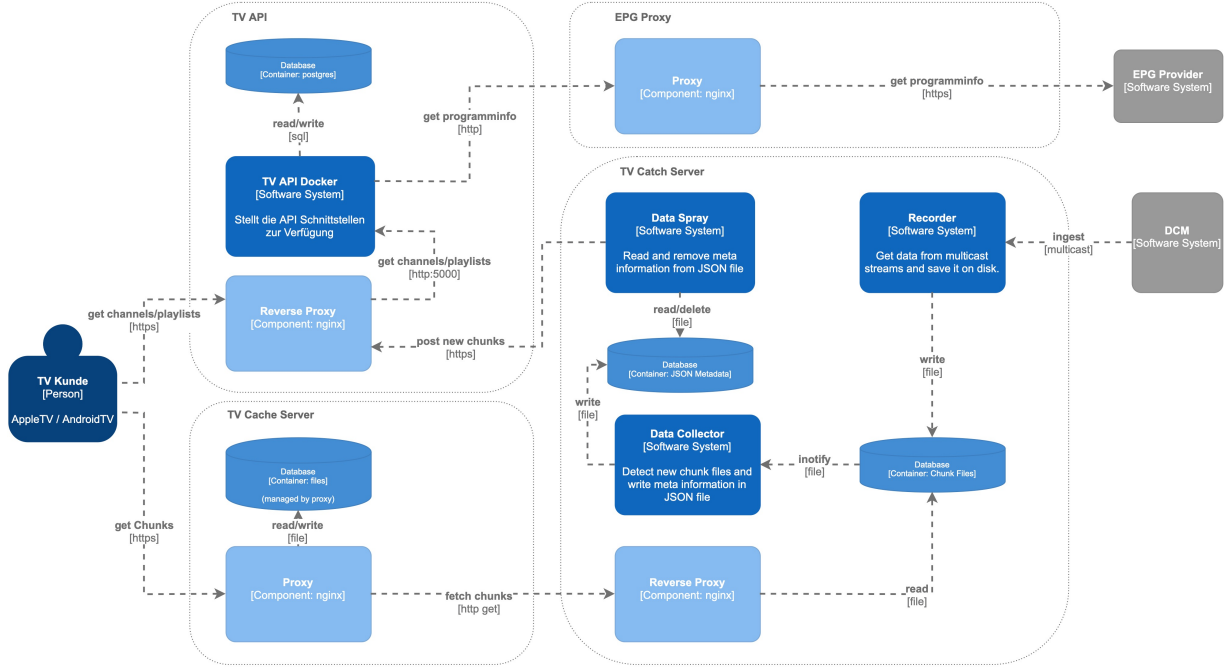
- Introduction
- Background & Theory
- **Methodology & Results**
- Discussion
- Outlook
- Demo
- Questions

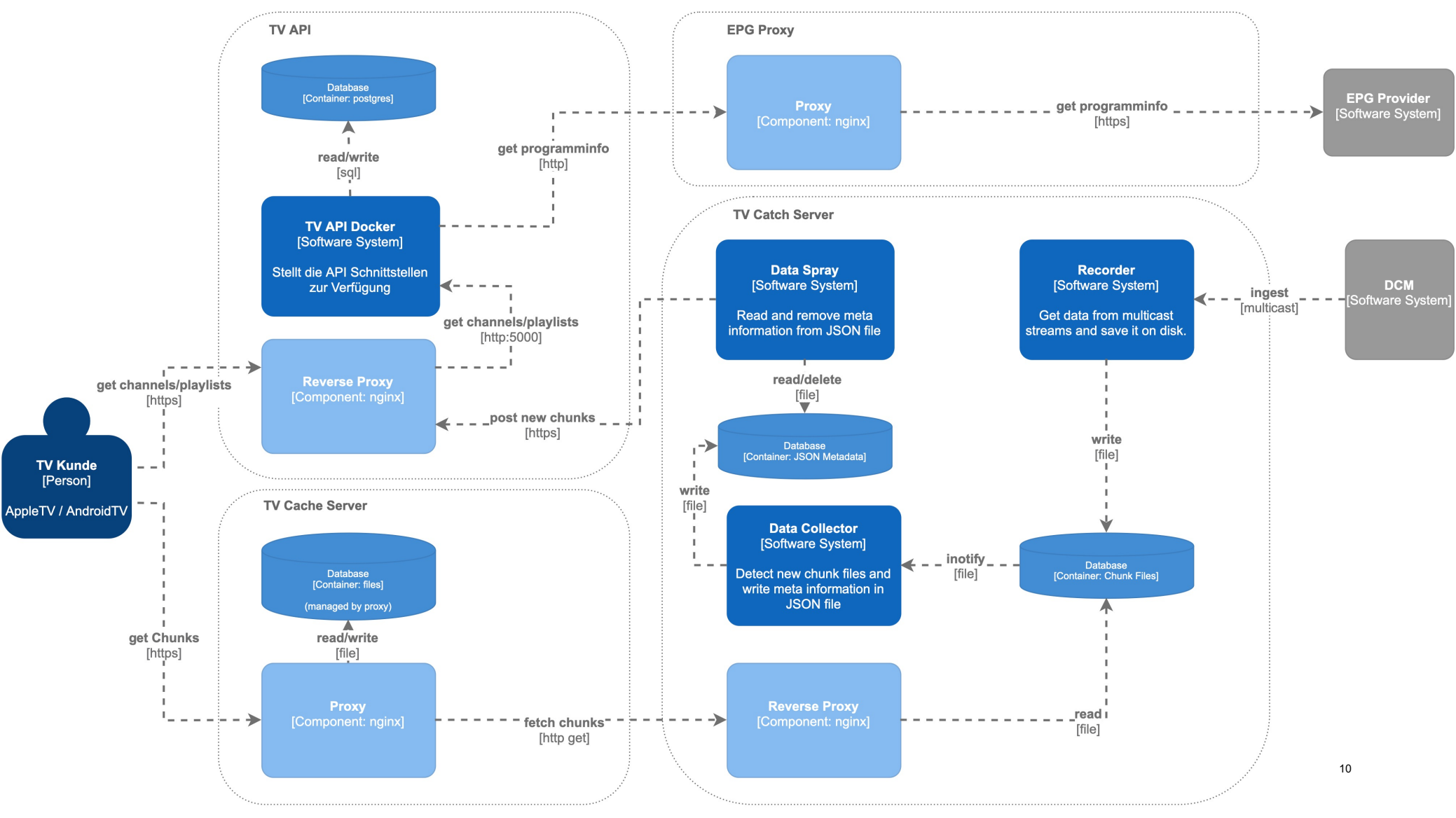
Methodology & Results

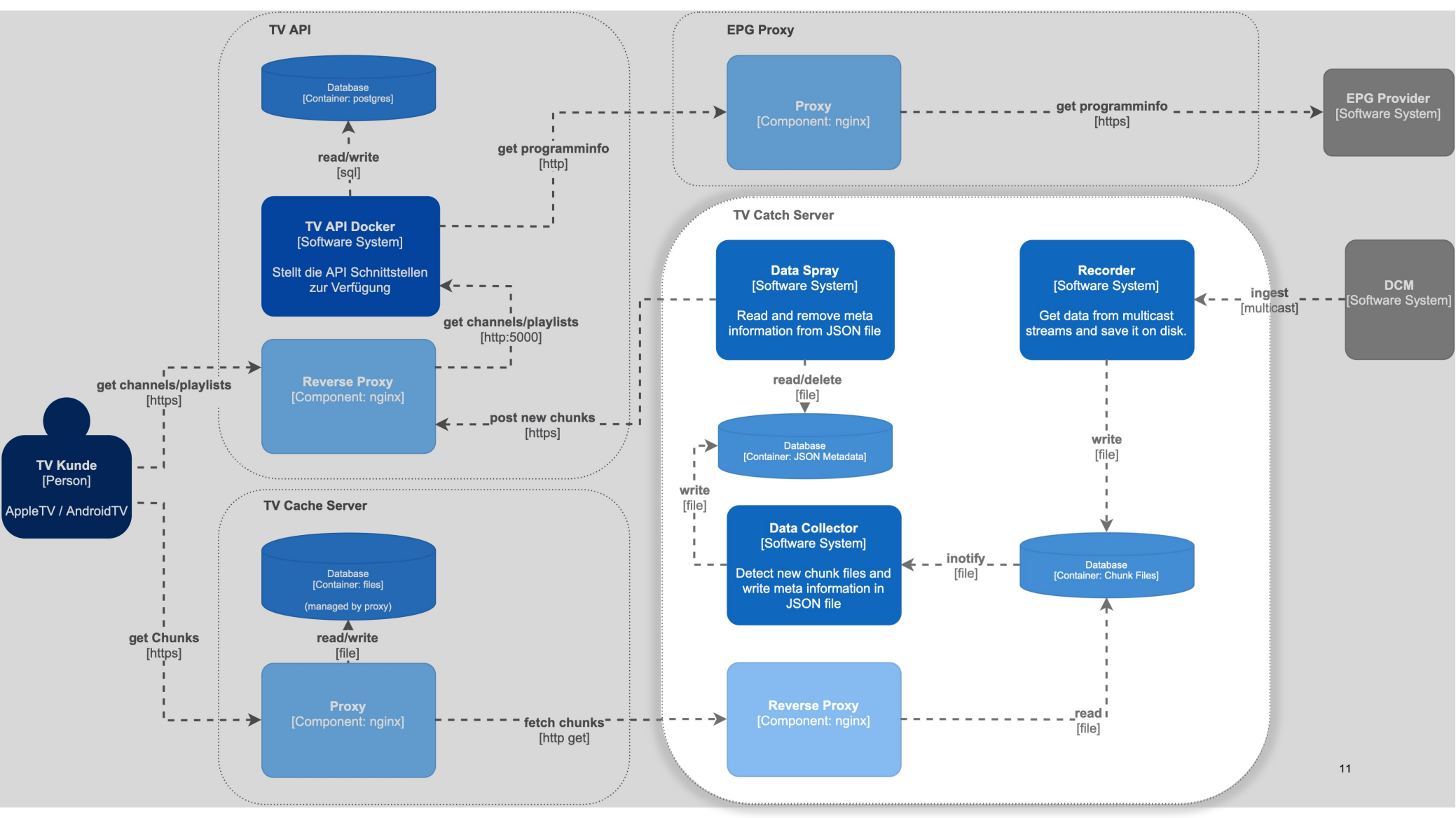
Zürcher Hochschule
für Angewandte Wissenschaften

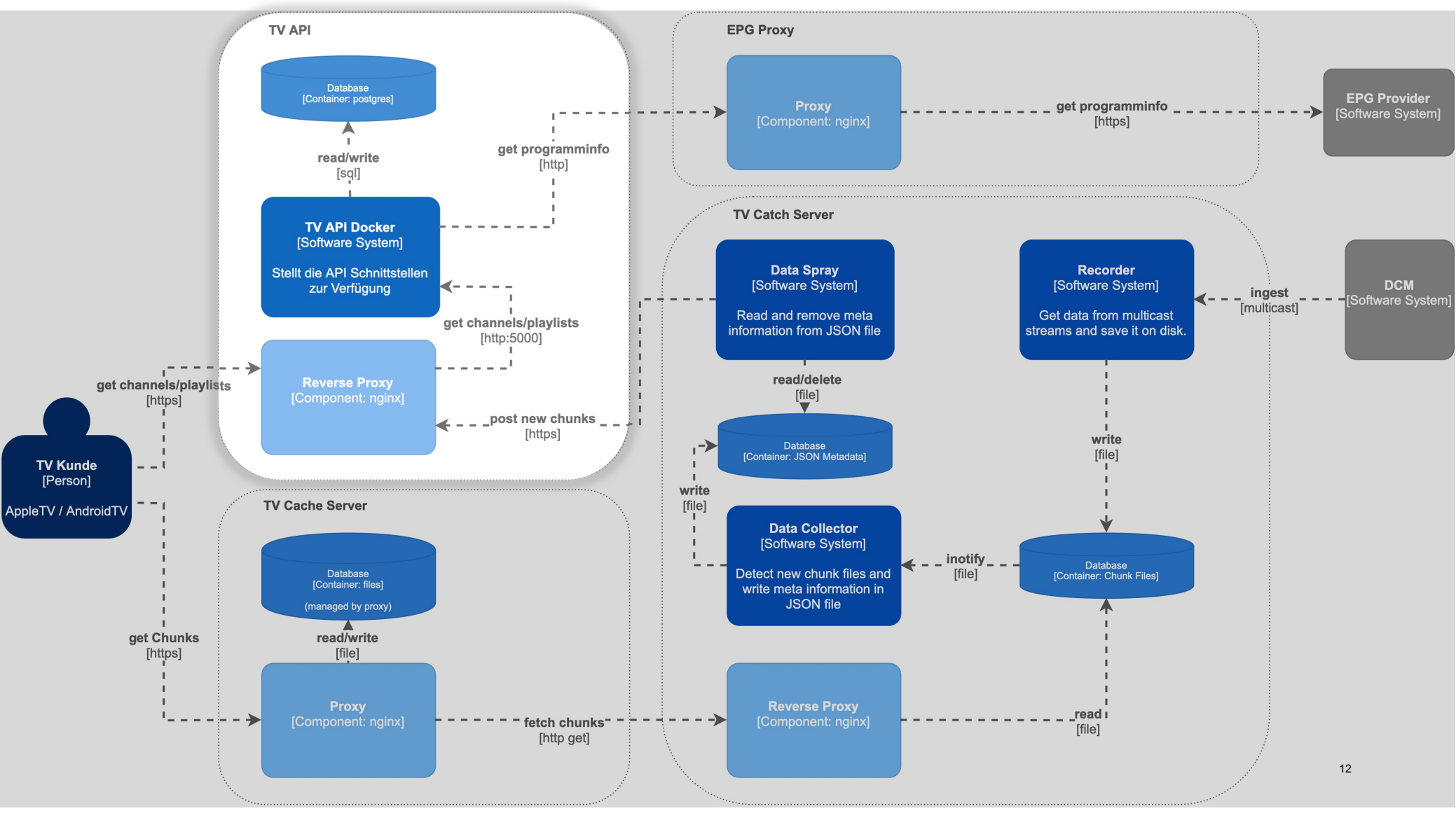
zh aw School of Engineering

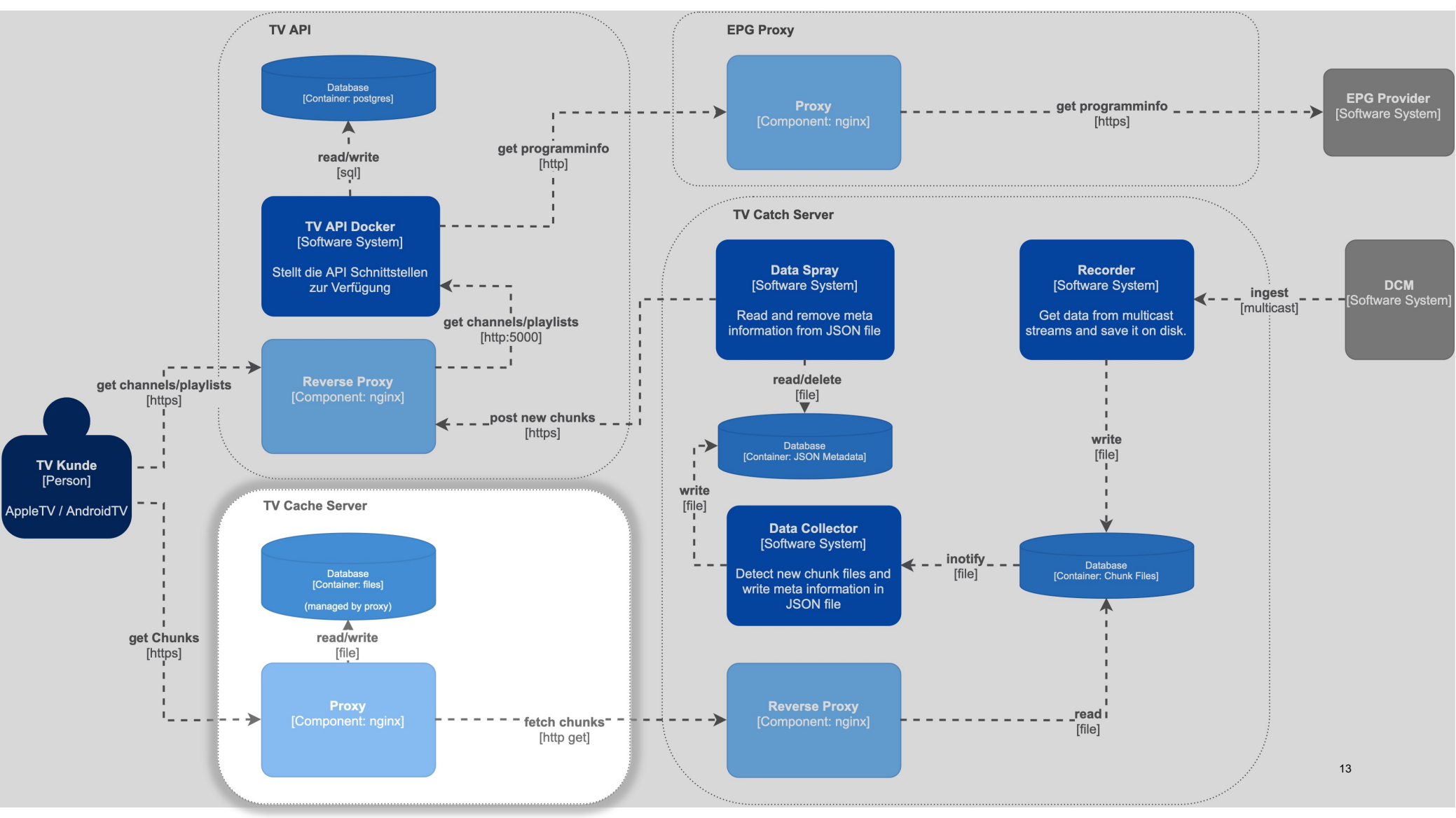
ANALYSIS CURRENT STATE

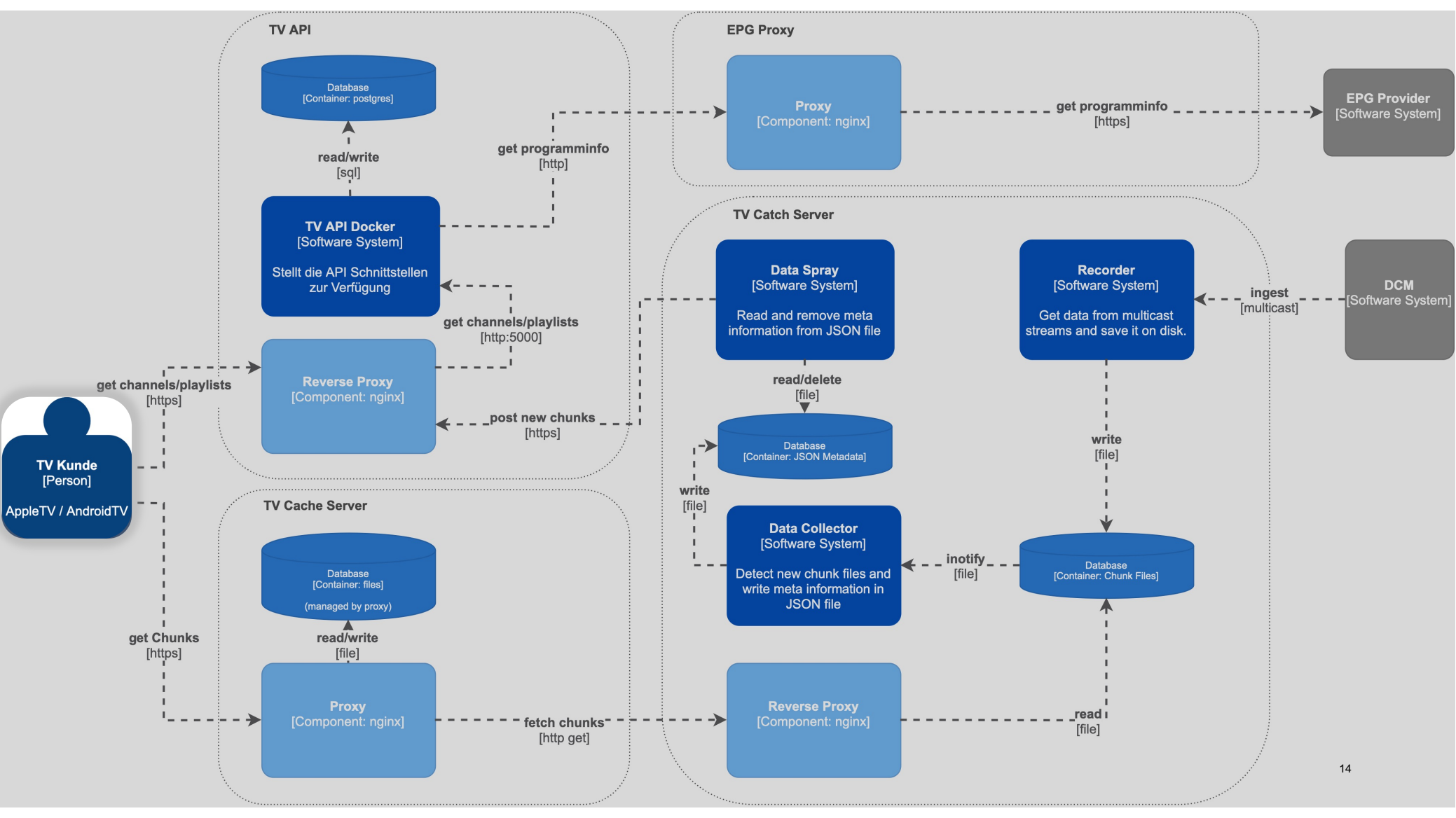



TV API
Database
[Container: postgres]
read/write
[sql]
TV API Docker
[Software System]
Stellt die API Schnittstellen zur Verfügung
get programminfo
[http]
get channels/playlists
[http:5000]
Reverse Proxy
[Component: nginx]
get channels/playlists
[https]
post new chunks
[https]
EPG Proxy
Proxy
[Component: nginx]
get programminfo
[https]
EPG Provider
[Software System]
TV Catch Server
Data Spray
[Software System]
Read and remove meta information from JSON file
read/delete
[file]
Database
[Container: JSON Metadata]
write
[file]
Recorder
[Software System]
Get data from multicast streams and save it on disk.
ingest
[multicast]
DCM
[Software System]
write
[file]
Data Collector
[Software System]
Detect new chunk files and write meta information in JSON file
inotify
[file]
Database
[Container: Chunk Files]
read
[file]
Reverse Proxy
[Component: nginx]
TV Kunde
[Person]
AppleTV / AndroidTV
TV Cache Server
Database
[Container: files]
(managed by proxy)
read/write
[file]
get Chunks
[https]
Proxy
[Component: nginx]
fetch chunks
[http get]


TV API
Database
[Container: postgres]
read/write
[sql]
TV API Docker
[Software System]
Stellt die API Schnittstellen zur Verfügung
get programminfo
[http]
get channels/playlists
[http:5000]
Reverse Proxy
[Component: nginx]
post new chunks
[https]
EPG Proxy
Proxy
[Component: nginx]
get programminfo
[https]
EPG Provider
[Software System]
TV Catch Server
Data Spray
[Software System]
Read and remove meta information from JSON file
read/delete
[file]
Database
[Container: JSON Metadata]
write
[file]
Data Collector
[Software System]
Detect new chunk files and write meta information in JSON file
inotify
[file]
Recorder
[Software System]
Get data from multicast streams and save it on disk.
ingest
[multicast]
DCM
[Software System]
write
[file]
Database
[Container: Chunk Files]
read
[file]
Reverse Proxy
[Component: nginx]
fetch chunks
[http get]
TV Cache Server
Database
[Container: files]
(managed by proxy)
read/write
[file]
Proxy
[Component: nginx]
get Chunks
[https]
get channels/playlists
[https]
TV Kunde
[Person]
AppleTV / AndroidTV


TV API
Database
[Container: postgres]
read/write
[sql]
get programminfo
[http]
TV API Docker
[Software System]
Stellt die API Schnittstellen zur Verfügung
get channels/playlists
[http:5000]
Reverse Proxy
[Component: nginx]
post new chunks
[https]
get channels/playlists
[https]
EPG Proxy
Proxy
[Component: nginx]
get programminfo
[https]
EPG Provider
[Software System]
TV Catch Server
Data Spray
[Software System]
Read and remove meta information from JSON file
read/delete
[file]
Database
[Container: JSON Metadata]
write
[file]
Recorder
[Software System]
Get data from multicast streams and save it on disk.
ingest
[multicast]
DCM
[Software System]
write
[file]
Data Collector
[Software System]
Detect new chunk files and write meta information in JSON file
inotify
[file]
Database
[Container: Chunk Files]
TV Kunde
[Person]
AppleTV / AndroidTV
TV Cache Server
Database
[Container: files]
(managed by proxy)
read/write
[file]
get Chunks
[https]
Proxy
[Component: nginx]
fetch chunks
[http get]
Reverse Proxy
[Component: nginx]
read
[file]


TV API
Database
[Container: postgres]
read/write
[sql]
TV API Docker
[Software System]
Stellt die API Schnittstellen zur Verfügung
get programminfo
[http]
get channels/playlists
[http:5000]
Reverse Proxy
[Component: nginx]
get channels/playlists
[https]
post new chunks
[https]
EPG Proxy
Proxy
[Component: nginx]
get programminfo
[https]
EPG Provider
[Software System]
TV Catch Server
Data Spray
[Software System]
Read and remove meta information from JSON file
read/delete
[file]
Database
[Container: JSON Metadata]
write
[file]
Recorder
[Software System]
Get data from multicast streams and save it on disk.
ingest
[multicast]
DCM
[Software System]
write
[file]
Data Collector
[Software System]
Detect new chunk files and write meta information in JSON file
inotify
[file]
Database
[Container: Chunk Files]
Reverse Proxy
[Component: nginx]
read
[file]
TV Kunde
[Person]
AppleTV / AndroidTV
TV Cache Server
Database
[Container: files]
(managed by proxy)
read/write
[file]
get Chunks
[https]
Proxy
[Component: nginx]
fetch chunks
[http get]

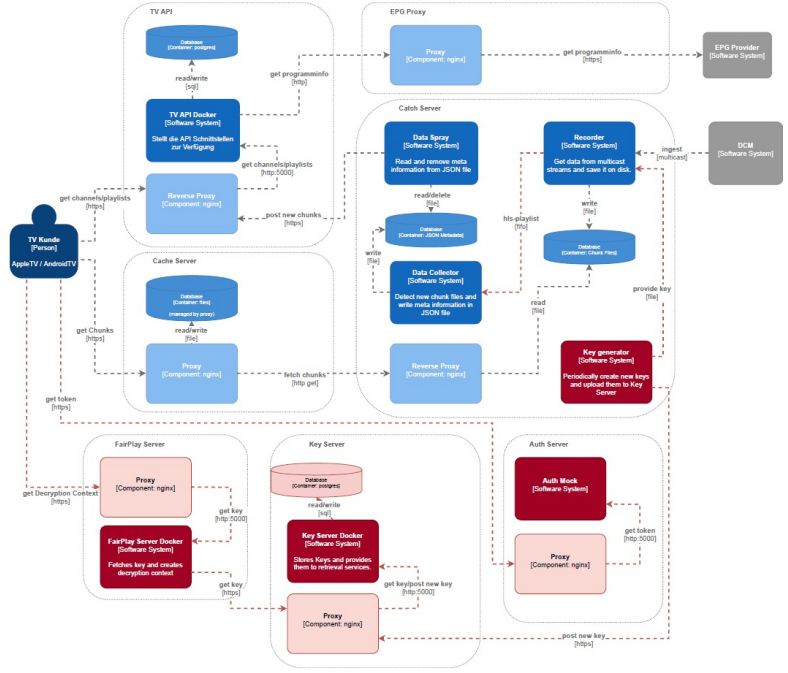

TV API
Database
[Container: postgres]
read/write
[sql]
TV API Docker
[Software System]
Stellt die API Schnittstellen zur Verfügung
get programminfo
[http]
get channels/playlists
[http:5000]
Reverse Proxy
[Component: nginx]
post new chunks
[https]
get channels/playlists
[https]
EPG Proxy
Proxy
[Component: nginx]
get programminfo
[https]
EPG Provider
[Software System]
TV Catch Server
Data Spray
[Software System]
Read and remove meta information from JSON file
read/delete
[file]
Database
[Container: JSON Metadata]
write
[file]
Recorder
[Software System]
Get data from multicast streams and save it on disk.
ingest
[multicast]
DCM
[Software System]
write
[file]
Data Collector
[Software System]
Detect new chunk files and write meta information in JSON file
inotify
[file]
Database
[Container: Chunk Files]
TV Kunde
[Person]
AppleTV / AndroidTV
TV Cache Server
Database
[Container: files]
(managed by proxy)
read/write
[file]
get Chunks
[https]
Proxy
[Component: nginx]
fetch chunks
[http get]
Reverse Proxy
[Component: nginx]
read
[file]

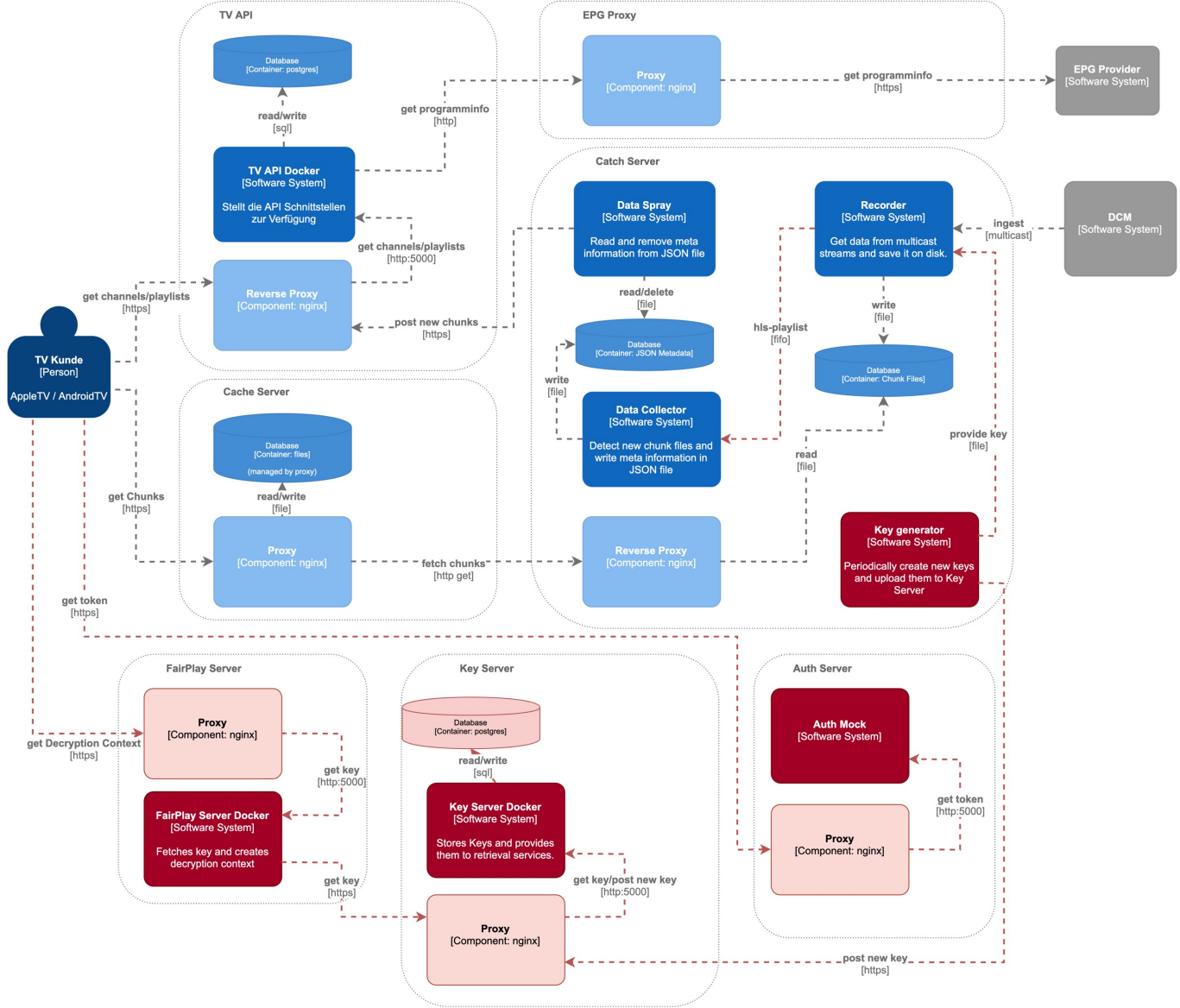
Methodology & Results

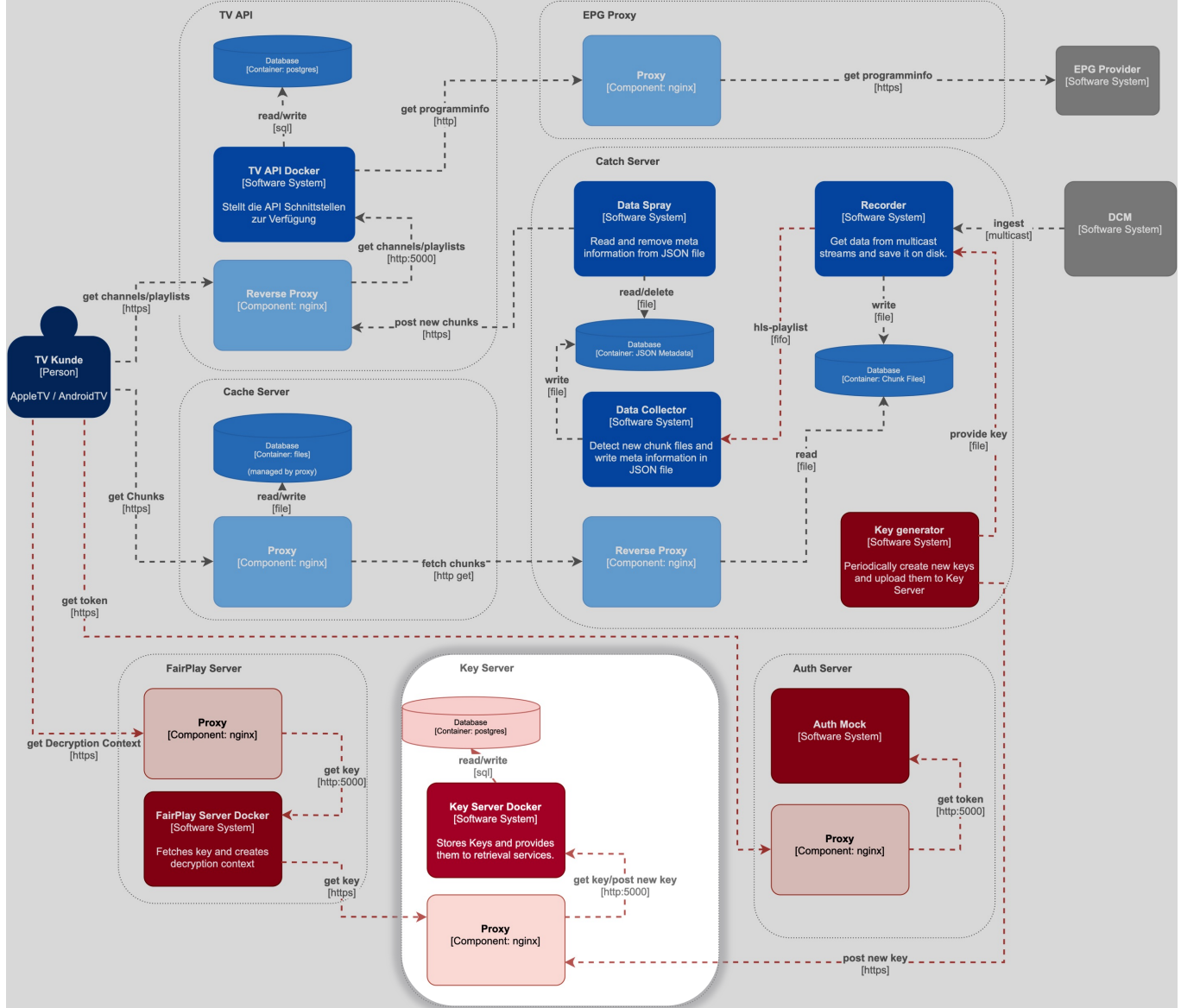
Zürcher Hochschule
für Angewandte Wissenschaften

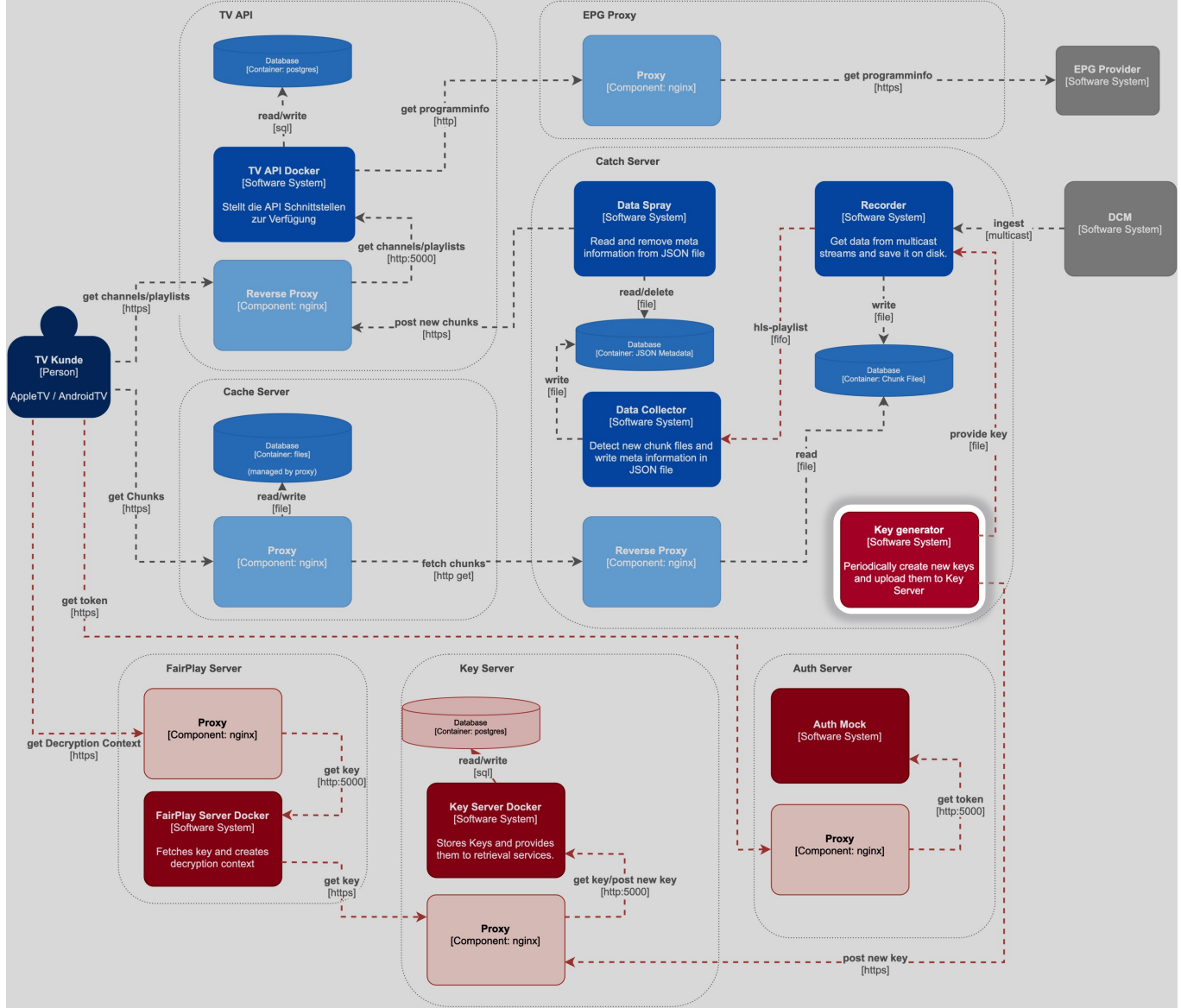
zh aw School of Engineering

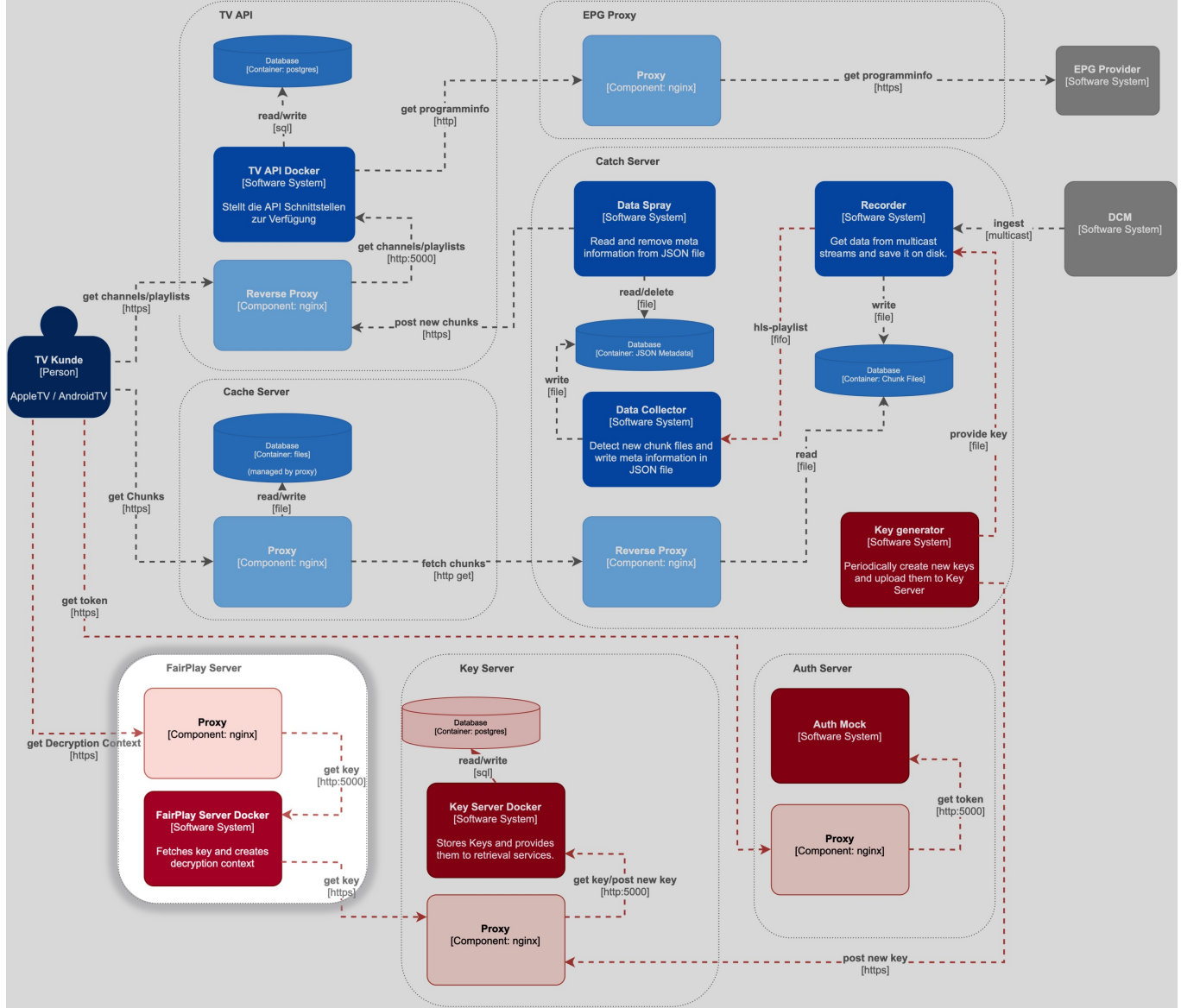
ARCHITECURE DRAFT

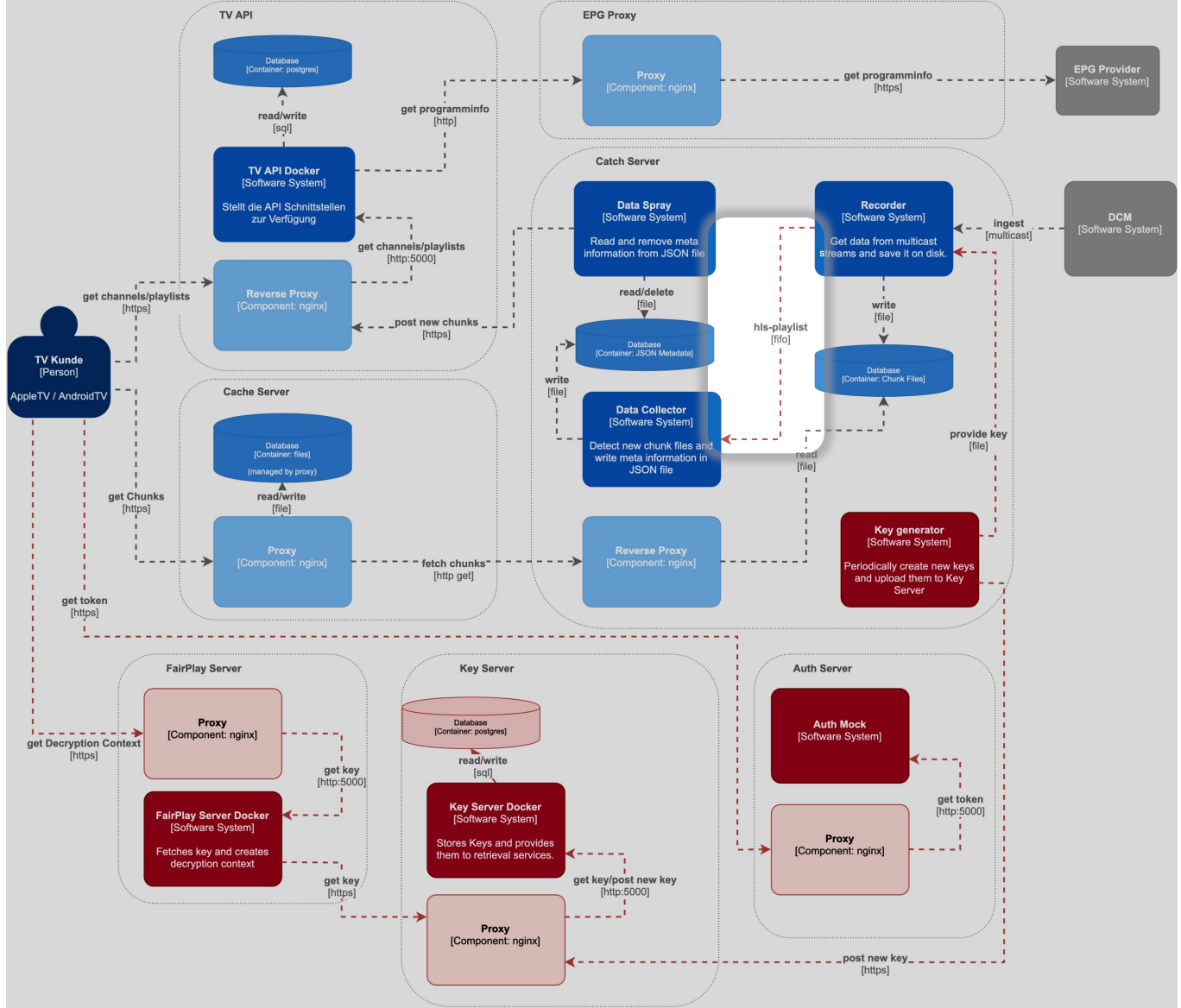


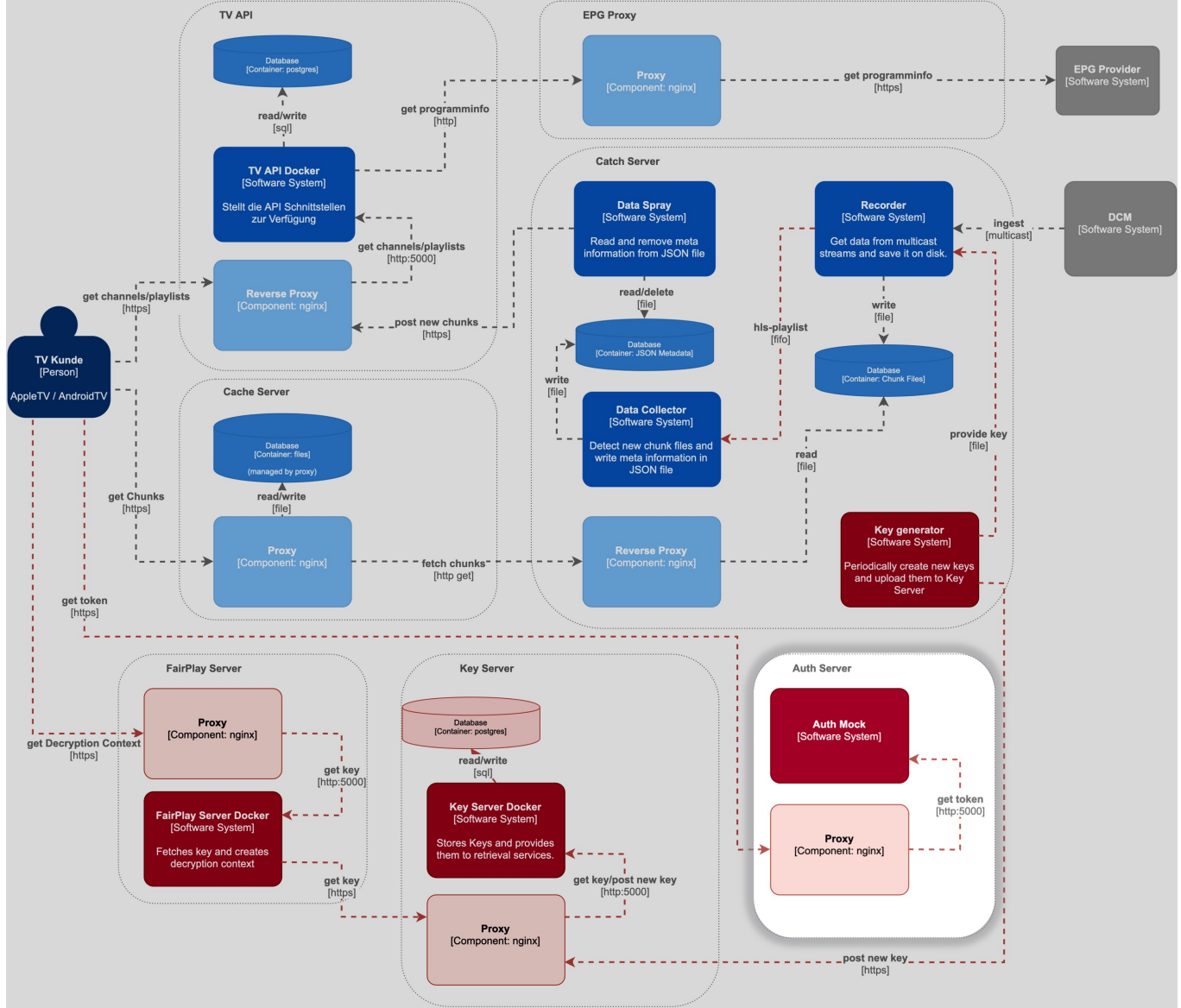

TV API
Database
[Container: postgres]
read/write
[sql]
get programminfo
[http]
TV API Docker
[Software System]
Stellt die API Schnittstellen zur Verfügung
get channels/playlists
[http:5000]
Reverse Proxy
[Component: nginx]
post new chunks
[https]
get channels/playlists
[https]
EPG Proxy
Proxy
[Component: nginx]
get programminfo
[https]
EPG Provider
[Software System]
Catch Server
Data Spray
[Software System]
Read and remove meta information from JSON file
read/delete
[file]
Database
[Container: JSON Metadata]
write
[file]
Data Collector
[Software System]
Detect new chunk files and write meta information in JSON file
Recorder
[Software System]
Get data from multicast streams and save it on disk.
ingest
[multicast]
DCM
[Software System]
write
[file]
Database
[Container: Chunk Files]
hls-playlist
[fifo]
read
[file]
provide key
[file]
Reverse Proxy
[Component: nginx]
Key generator
[Software System]
Periodically create new keys and upload them to Key Server
TV Kunde
[Person]
AppleTV / AndroidTV
Cache Server
Database
[Container: files]
(managed by proxy)
read/write
[file]
get Chunks
[https]
Proxy
[Component: nginx]
fetch chunks
[http get]
get token
[https]
FairPlay Server
Proxy
[Component: nginx]
get Decryption Context
[https]
get key
[http:5000]
FairPlay Server Docker
[Software System]
Fetches key and creates decryption context
get key
[https]
Key Server
Database
[Container: postgres]
read/write
[sql]
Key Server Docker
[Software System]
Stores Keys and provides them to retrieval services.
get key/post new key
[http:5000]
Proxy
[Component: nginx]
Auth Server
Auth Mock
[Software System]
get token
[http:5000]
Proxy
[Component: nginx]
post new key
[https]


TV API
Database
[Container: postgres]
read/write
[sql]
TV API Docker
[Software System]
Stellt die API Schnittstellen zur Verfügung
get programminfo
[http]
get channels/playlists
[http:5000]
Reverse Proxy
[Component: nginx]
post new chunks
[https]
get channels/playlists
[https]
EPG Proxy
Proxy
[Component: nginx]
get programminfo
[https]
EPG Provider
[Software System]
Catch Server
Data Spray
[Software System]
Read and remove meta information from JSON file
read/delete
[file]
Database
[Container: JSON Metadata]
write
[file]
Data Collector
[Software System]
Detect new chunk files and write meta information in JSON file
Recorder
[Software System]
Get data from multicast streams and save it on disk.
ingest
[multicast]
DCM
[Software System]
hls-playlist
[fifo]
write
[file]
Database
[Container: Chunk Files]
read
[file]
provide key
[file]
Reverse Proxy
[Component: nginx]
Key generator
[Software System]
Periodically create new keys and upload them to Key Server
TV Kunde
[Person]
AppleTV / AndroidTV
Cache Server
Database
[Container: files]
(managed by proxy)
read/write
[file]
get Chunks
[https]
Proxy
[Component: nginx]
fetch chunks
[http get]
get token
[https]
FairPlay Server
Proxy
[Component: nginx]
get Decryption Context
[https]
get key
[http:5000]
FairPlay Server Docker
[Software System]
Fetches key and creates decryption context
get key
[https]
Key Server
Database
[Container: postgres]
read/write
[sql]
Key Server Docker
[Software System]
Stores Keys and provides them to retrieval services.
get key/post new key
[http:5000]
Proxy
[Component: nginx]
post new key
[https]
Auth Server
Auth Mock
[Software System]
get token
[http:5000]
Proxy
[Component: nginx]


TV API
Database
[Container: postgres]
read/write
[sql]
TV API Docker
[Software System]
Stellt die API Schnittstellen zur Verfügung
get programminfo
[http]
get channels/playlists
[http:5000]
Reverse Proxy
[Component: nginx]
post new chunks
[https]
get channels/playlists
[https]
EPG Proxy
Proxy
[Component: nginx]
get programminfo
[https]
EPG Provider
[Software System]
Catch Server
Data Spray
[Software System]
Read and remove meta information from JSON file
read/delete
[file]
Database
[Container: JSON Metadata]
write
[file]
Data Collector
[Software System]
Detect new chunk files and write meta information in JSON file
Recorder
[Software System]
Get data from multicast streams and save it on disk.
ingest
[multicast]
DCM
[Software System]
hls-playlist
[fifo]
write
[file]
Database
[Container: Chunk Files]
read
[file]
provide key
[file]
Reverse Proxy
[Component: nginx]
Key generator
[Software System]
Periodically create new keys and upload them to Key Server
TV Kunde
[Person]
AppleTV / AndroidTV
Cache Server
Database
[Container: files]
(managed by proxy)
read/write
[file]
get Chunks
[https]
Proxy
[Component: nginx]
fetch chunks
[http get]
get token
[https]
FairPlay Server
Proxy
[Component: nginx]
get Decryption Context
[https]
get key
[http:5000]
FairPlay Server Docker
[Software System]
Fetches key and creates decryption context
get key
[https]
Key Server
Database
[Container: postgres]
read/write
[sql]
Key Server Docker
[Software System]
Stores Keys and provides them to retrieval services.
get key/post new key
[http:5000]
Proxy
[Component: nginx]
Auth Server
Auth Mock
[Software System]
get token
[http:5000]
Proxy
[Component: nginx]
post new key
[https]


TV API
Database
[Container: postgres]
read/write
[sql]
TV API Docker
[Software System]
Stellt die API Schnittstellen zur Verfügung
get programminfo
[http]
get channels/playlists
[http:5000]
Reverse Proxy
[Component: nginx]
post new chunks
[https]
EPG Proxy
Proxy
[Component: nginx]
get programminfo
[https]
EPG Provider
[Software System]
Catch Server
Data Spray
[Software System]
Read and remove meta information from JSON file
read/delete
[file]
Database
[Container: JSON Metadata]
write
[file]
Data Collector
[Software System]
Detect new chunk files and write meta information in JSON file
Recorder
[Software System]
Get data from multicast streams and save it on disk.
ingest
[multicast]
DCM
[Software System]
hls-playlist
[fifo]
write
[file]
Database
[Container: Chunk Files]
read
[file]
provide key
[file]
Reverse Proxy
[Component: nginx]
Key generator
[Software System]
Periodically create new keys and upload them to Key Server
get channels/playlists
[https]
TV Kunde
[Person]
AppleTV / AndroidTV
get Chunks
[https]
get token
[https]
Cache Server
Database
[Container: files]
(managed by proxy)
read/write
[file]
Proxy
[Component: nginx]
fetch chunks
[http get]
FairPlay Server
Proxy
[Component: nginx]
get Decryption Context
[https]
get key
[http:5000]
FairPlay Server Docker
[Software System]
Fetches key and creates decryption context
get key
[https]
Key Server
Database
[Container: postgres]
read/write
[sql]
Key Server Docker
[Software System]
Stores Keys and provides them to retrieval services.
get key/post new key
[http:5000]
Proxy
[Component: nginx]
Auth Server
Auth Mock
[Software System]
get token
[http:5000]
Proxy
[Component: nginx]
post new key
[https]


TV API
Database
[Container: postgres]
read/write
[sql]
TV API Docker
[Software System]
Stellt die API Schnittstellen zur Verfügung
get programminfo
[http]
get channels/playlists
[http:5000]
Reverse Proxy
[Component: nginx]
post new chunks
[https]
EPG Proxy
Proxy
[Component: nginx]
get programminfo
[https]
EPG Provider
[Software System]
Catch Server
Data Spray
[Software System]
Read and remove meta information from JSON file
read/delete
[file]
Database
[Container: JSON Metadata]
write
[file]
Data Collector
[Software System]
Detect new chunk files and write meta information in JSON file
hls-playlist
[fifo]
Recorder
[Software System]
Get data from multicast streams and save it on disk.
write
[file]
Database
[Container: Chunk Files]
read
[file]
ingest
[multicast]
DCM
[Software System]
provide key
[file]
Reverse Proxy
[Component: nginx]
Key generator
[Software System]
Periodically create new keys and upload them to Key Server
get channels/playlists
[https]
TV Kunde
[Person]
AppleTV / AndroidTV
Cache Server
Database
[Container: files]
(managed by proxy)
read/write
[file]
get Chunks
[https]
Proxy
[Component: nginx]
fetch chunks
[http get]
get token
[https]
FairPlay Server
Proxy
[Component: nginx]
get Decryption Context
[https]
get key
[http:5000]
FairPlay Server Docker
[Software System]
Fetches key and creates decryption context
get key
[https]
Key Server
Database
[Container: postgres]
read/write
[sql]
Key Server Docker
[Software System]
Stores Keys and provides them to retrieval services.
get key/post new key
[http:5000]
Proxy
[Component: nginx]
Auth Server
Auth Mock
[Software System]
get token
[http:5000]
Proxy
[Component: nginx]
post new key
[https]


TV API
Database
[Container: postgres]
read/write
[sql]
TV API Docker
[Software System]
Stellt die API Schnittstellen zur Verfügung
get programminfo
[http]
get channels/playlists
[http:5000]
Reverse Proxy
[Component: nginx]
post new chunks
[https]
get channels/playlists
[https]
EPG Proxy
Proxy
[Component: nginx]
get programminfo
[https]
EPG Provider
[Software System]
Catch Server
Data Spray
[Software System]
Read and remove meta information from JSON file
read/delete
[file]
Database
[Container: JSON Metadata]
write
[file]
Data Collector
[Software System]
Detect new chunk files and write meta information in JSON file
Recorder
[Software System]
Get data from multicast streams and save it on disk.
ingest
[multicast]
DCM
[Software System]
hls-playlist
[fifo]
write
[file]
Database
[Container: Chunk Files]
read
[file]
provide key
[file]
Reverse Proxy
[Component: nginx]
Key generator
[Software System]
Periodically create new keys and upload them to Key Server
TV Kunde
[Person]
AppleTV / AndroidTV
Cache Server
Database
[Container: files]
(managed by proxy)
read/write
[file]
get Chunks
[https]
Proxy
[Component: nginx]
fetch chunks
[http get]
get token
[https]
FairPlay Server
Proxy
[Component: nginx]
get Decryption Context
[https]
get key
[http:5000]
FairPlay Server Docker
[Software System]
Fetches key and creates decryption context
get key
[https]
Key Server
Database
[Container: postgres]
read/write
[sql]
Key Server Docker
[Software System]
Stores Keys and provides them to retrieval services.
get key/post new key
[http:5000]
Proxy
[Component: nginx]
Auth Server
Auth Mock
[Software System]
get token
[http:5000]
Proxy
[Component: nginx]
post new key
[https]

Methodology & Results

Zürcher Hochschule
für Angewandte Wissenschaften
zh aw School of Engineering

PROTOTYPE

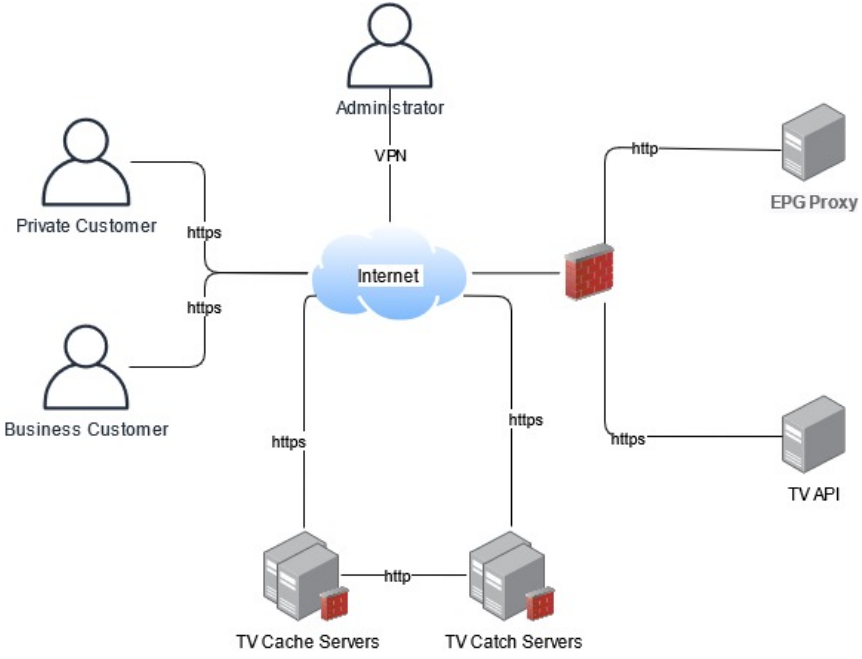
- Key Server, Key Generator, FairPlay Streaming Server, Demo App work as planned
- Auth Server out of scope
- Extension of existing system didn’t go as well
  - TV API easy
  - Encryption worked too
  - But FairPlay requires SAMPLE-AES, which ffmpeg can’t do
  - Attempts to replace ffmpeg with shaka packager (Google) failed due to existing architecture

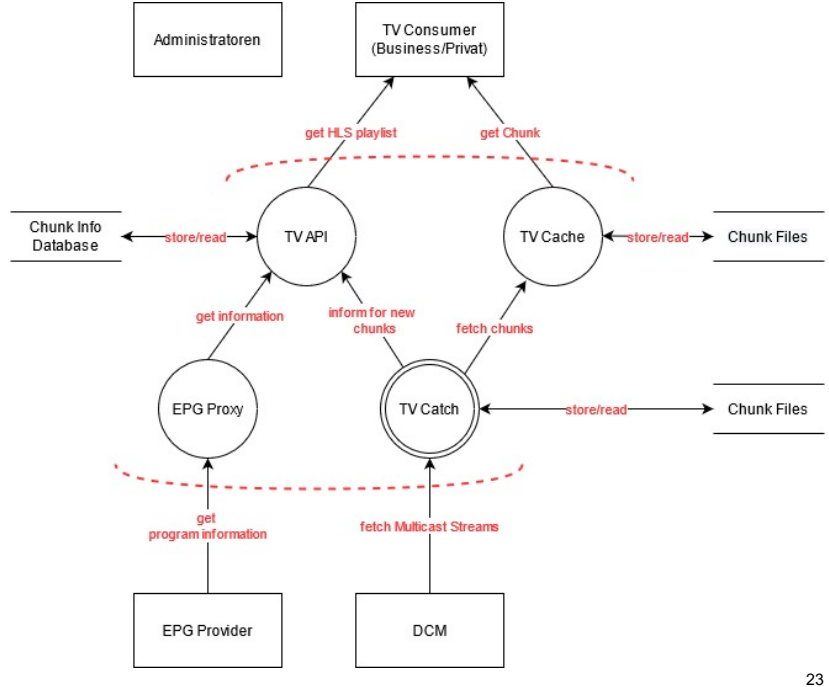
Methodology & Results

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

BASIC THREAT ANALYSIS OF THE CURRENT SITUATION

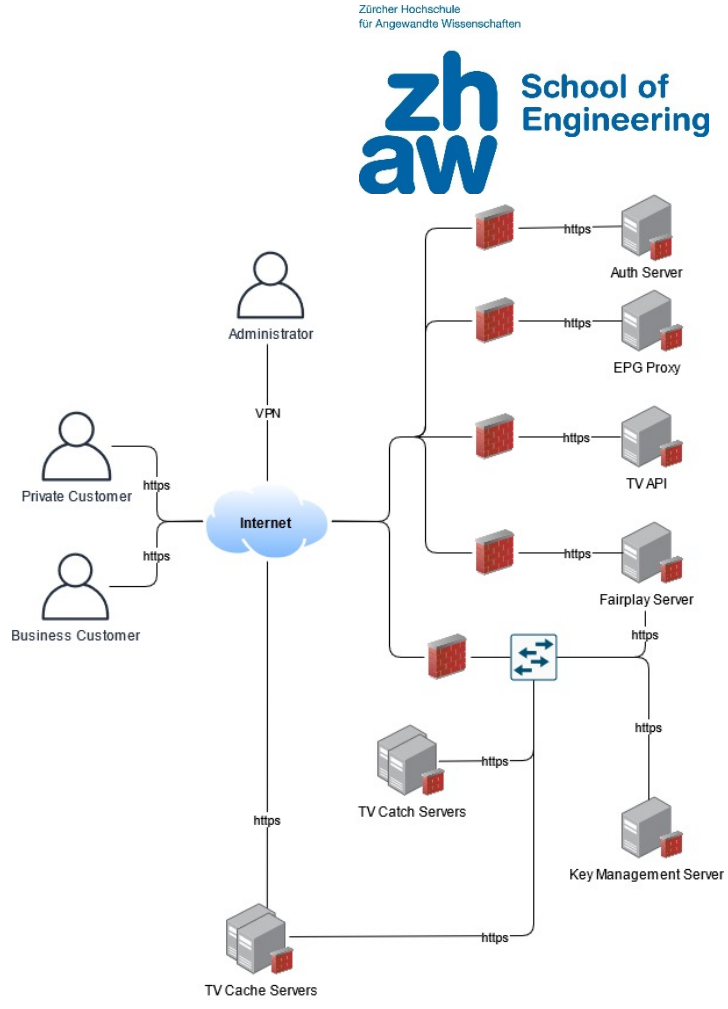




Methodology & Results

THREAT ANALYSIS (1/2)

- Network traffic over HTTPS
- Catch and Key Server in separate network, no access from outside
- Fairplay and Cache Server access to this network
- Nftables firewall for every server



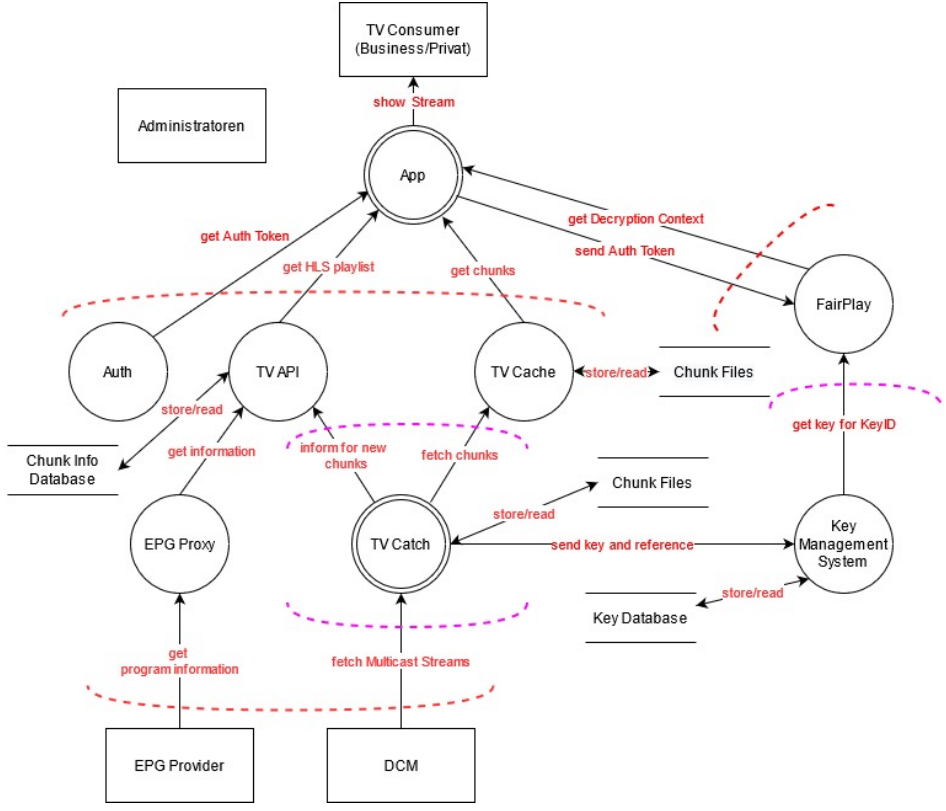
Methodology & Results

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

THREAT ANALYSIS (2/2)

- Can only view stream with app from TV provider
- Pink dashed line is lower area of confidence, only for key exchange



Methodology & Results

Zürcher Hochschule
für Angewandte Wissenschaften
zh aw School of Engineering

THREATS

- 112 threats
- Four main attacks:
  - Fake streams → influence customers
  - Unauthorised persons view stream → miss out on revenue
  - DoS attacks → components can fail, partital failure of television broadcaster and gap in recording
  - Gain admin access

MITIGATION

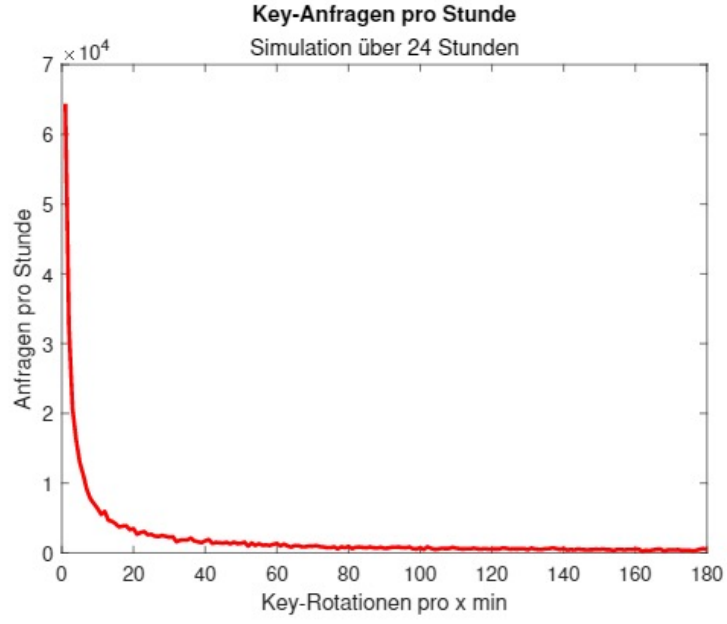
- Redundancy
- Administrator should only have the access he/she needs
- No shared accounts
- Login attempts limited
- Firewalls with white- and blacklisting

Methodology & Results

Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering

Analyzing the key rotation:

- Simulation with matlab
- Maximum growth of 3 customers per minute (No shrinkage)
- Curve is inversely proportional
- Large inquiries despite small number of customers



Methodology & Results

Zürcher Hochschule
für Angewandte Wissenschaften
zh aw School of Engineering

Biggest Risks:

- Sharing the stream via VPN
- Attacker obtains or forges a token
- Availability of critical systems
- Storage systems
- Customization of code or data
- Vulnerability through administrators
- Fake the app

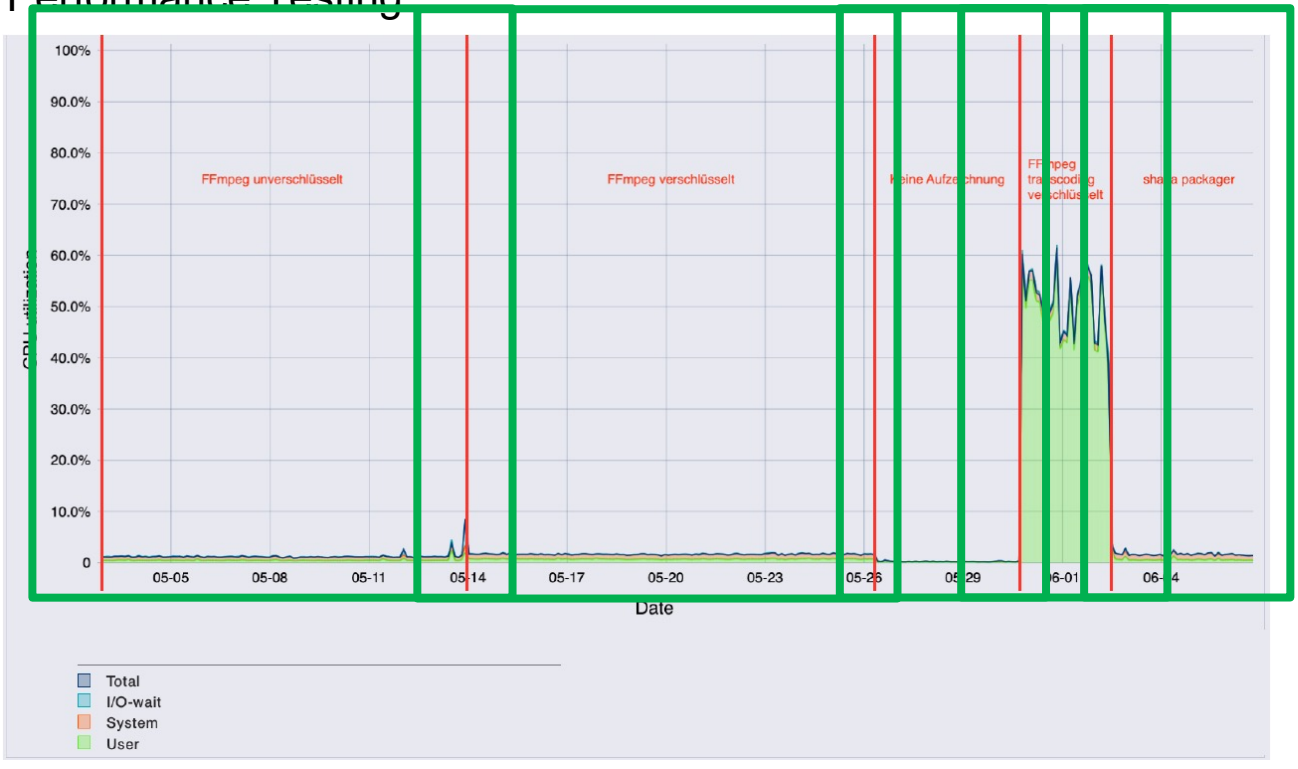
	Schaden		
Ausnutzung	Low	Medium	High
High	Low	Medium	High
Medium	Low	Medium	Medium
Low	Low	Low	Low

Methodology & Results

Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

Performance Testing



- Only 1 channel

Server-HW:

- Intel Xeon Processor E3-1265L v2
- 32 GB RAM
- 1TB, 7200rpm harddisk

Index

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

- Introduction
- Background & Theory
- Methodology & Results
- **Discussion**
- Outlook
- Demo
- Questions

Discussion

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

THEORY

- Very little literature
- A lot of work for small and independent providers looking to implement DRM
- For full use case, Multi-DRM is necessary

Discussion

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

ARCHITECTURE/PROTOTYPE

- Architectural draft adequate
- Potential for improvement
  - E.g. Single point of failure with Key Server DB
  - E.g. Preloading of keys not trivial

Discussion

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

THREAT ANALYSIS

- Do not trust own systems, always authenticate
- Do not blindly trust employee logins
- Distributed system not only easier to maintain, but also better for security
- Security is very deployment specific
- Basic best practices apply

Discussion

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

EVALUATION

- Key rotation interval > 20 minutes
- Encryption overhead neglectable

Index

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

- Introduction
- Background & Theory
- Methodology & Results
- Discussion
- **Outlook**
- Demo
- Questions

Outlook

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

- Further development of prototype
- Extension to Multi-DRM System
- Optimization of Key Exchange
- Authentication System
- Redundancy and Cloud-Nativeness
- Usage of DRM options

Demo Time!

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

Questions/Discussion

Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering