

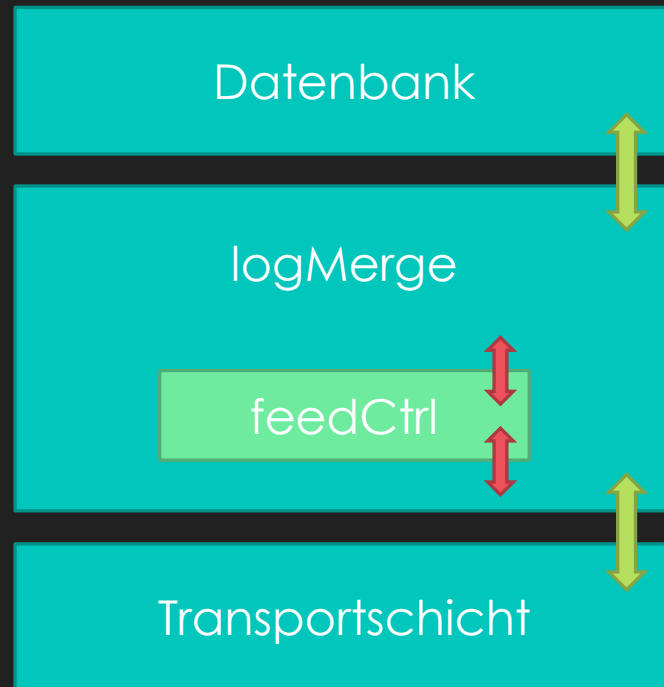
# Gruppe 4 - LogMerge

Günes Aydin, Nikodem Kernbach und Joey Zraggen

# Unser Auftrag



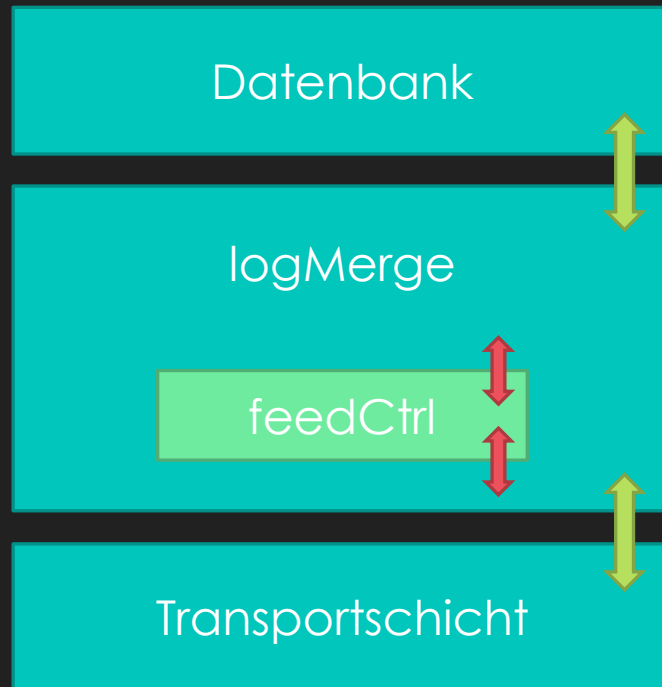
# Beispielablauf



Fall 1: Pakete kommen an  
"Von unten nach oben"

1. Extrahieren der .pcap-Files
2. Abgleich Stand Datenbank
3. Verifizierung der Signaturen / Hashes
4. Schreiben in Datenbank

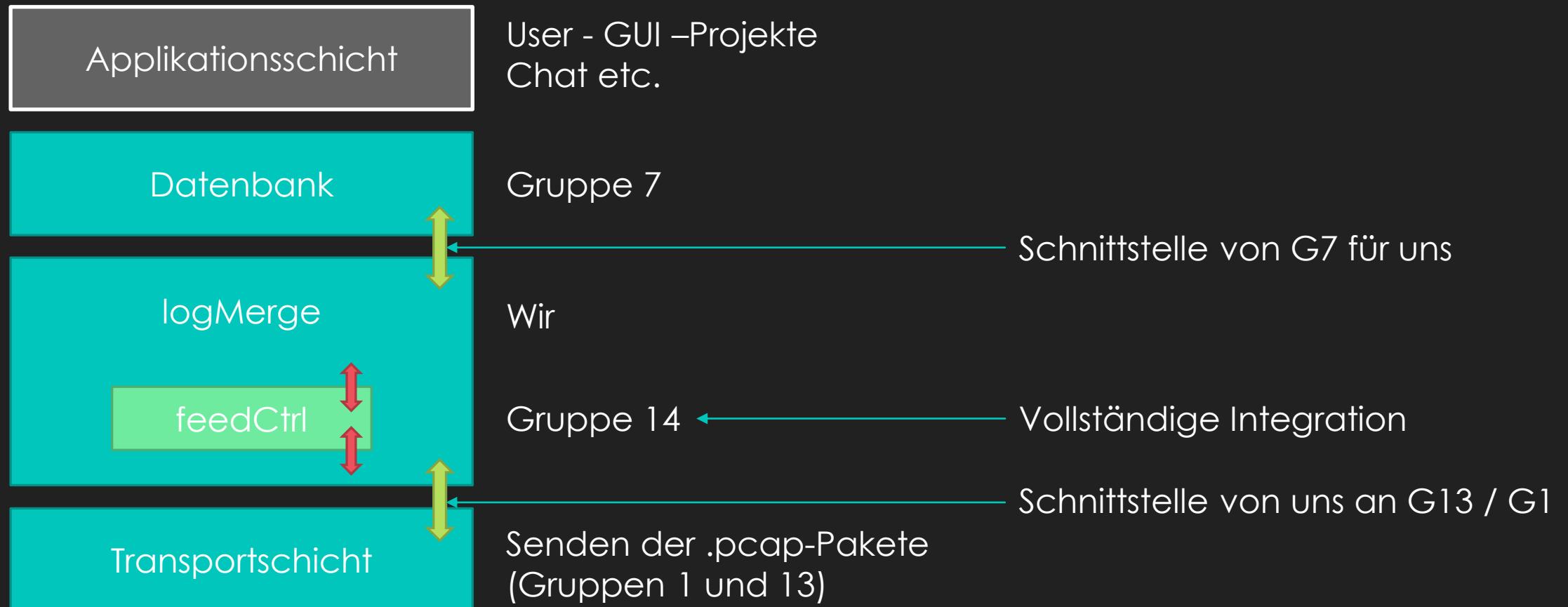
# Beispielablauf



Fall 2: Daten werden abgerufen  
"Von oben nach unten"

1. Transportschicht ruft uns auf
2. Ermitteln der zu sendenden Pakete
3. Events in .pcap-Files schreiben
4. Senden der Daten über Transportgruppe

# Schnittstellen



# Schnittstellen

- Zur Gruppe 7:
  - `get_current_seq_no(feed_id): seq_no`
  - `get_event(feed_id, seq_no): event`
  - `get_current_event(feed_id): event`
  - `add_event(feed_id, seq_no, event): void`
  - `get_secret_hmac_key(feed_id): secret_key`
- Zur Gruppe 13 / 1:
  - `export_logs(path_to_pcap_folder, dict_feed_id_current_seq_no, maximum_events_per_feed_id)`
  - `import_logs(self, paths_of_pcap_files)`
- Zur Gruppe 14:
  - Absprache nötig

# Aktueller Stand

- Schnittstellen Gruppe 7 logStore abgeklärt
- Import / Export von Events
- PCAP lesen / schreiben
- Unterstützung folgender Signaturen:
  - ED25519
  - HMAC\_SHA256
  - Weitere bei Bedarf
- Unterstützung von SHA256 Hashes (weitere bei Bedarf)

Vorzeigen **LIVE**

# Create Event Beispiel



# Weiters Vorgehen (was noch fehlt)

- Integration Gruppe 14 feedCtrl -> Filterfunktion
- Absprache G13 / 1
- Alles testen
- Evtl. Schnittstellen nachjustieren G7

# Eigenes TODO:

- Repository aufräumen
- Code strukturieren / kommentieren
- Integration Gruppe 14 FeedCtrl
- Erneutes Meeting mit Gruppe 13 SneakerNet (und 1 devtodev)