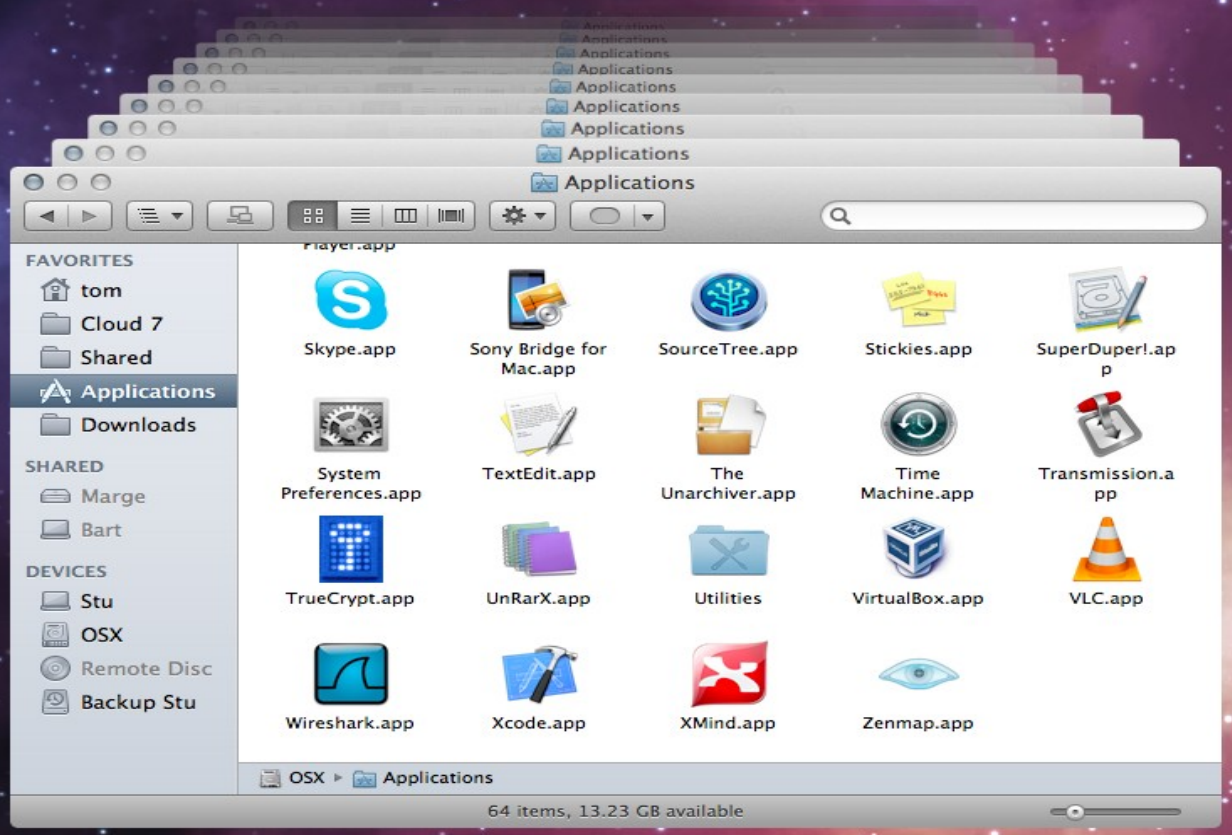
Timeline Forensics



T. Hospenthal

HSLU T&A
Diplomarbeit MAS
IT Network Manager

1929 René Margritte: „La trahison des images“ („Der Verrat der Bilder“)

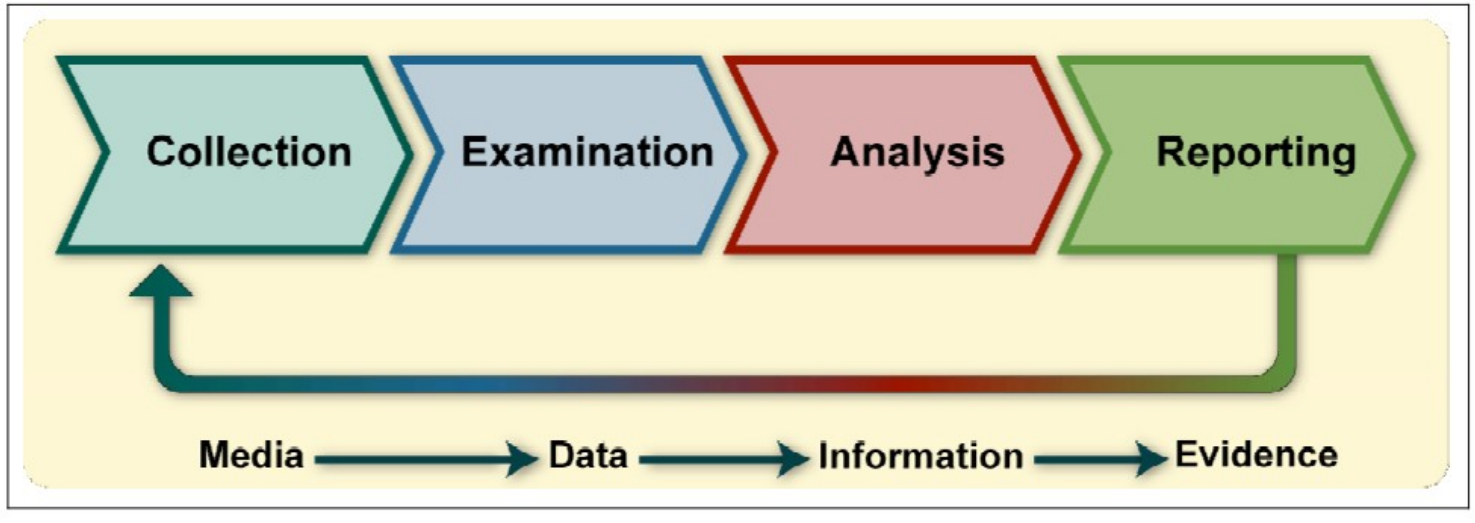
Digitale Forensik

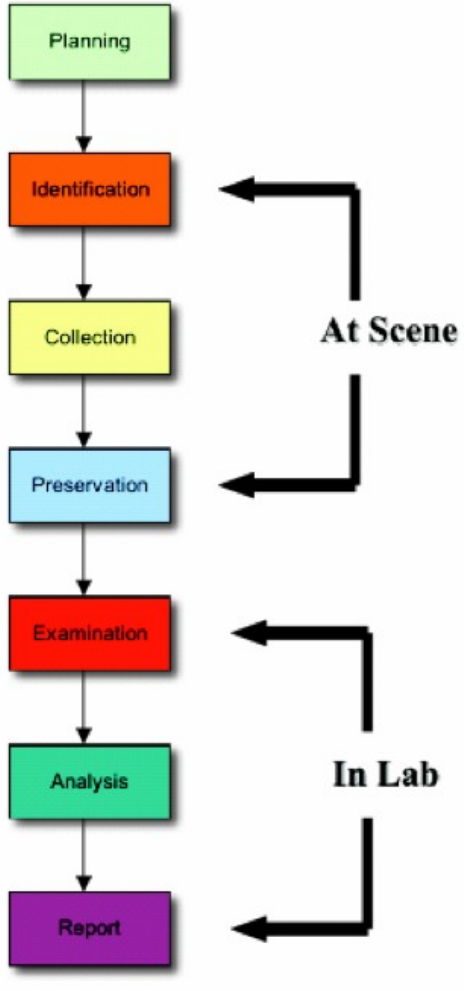
- Digitale Forensik als forensischen Wissenschaft
- Fachkundige Beurteilung eines Sachverhalts vor Gericht
- Nachvollziehbare und reproduzierbare Methodik
- Anwendung im Bereich der Computerkriminalität
- Anwendung vermehrt bei herkömmlicher Kriminalität

Der digital-forensische Prozess

Herausforderungen der digital-forensischen Untersuchung:

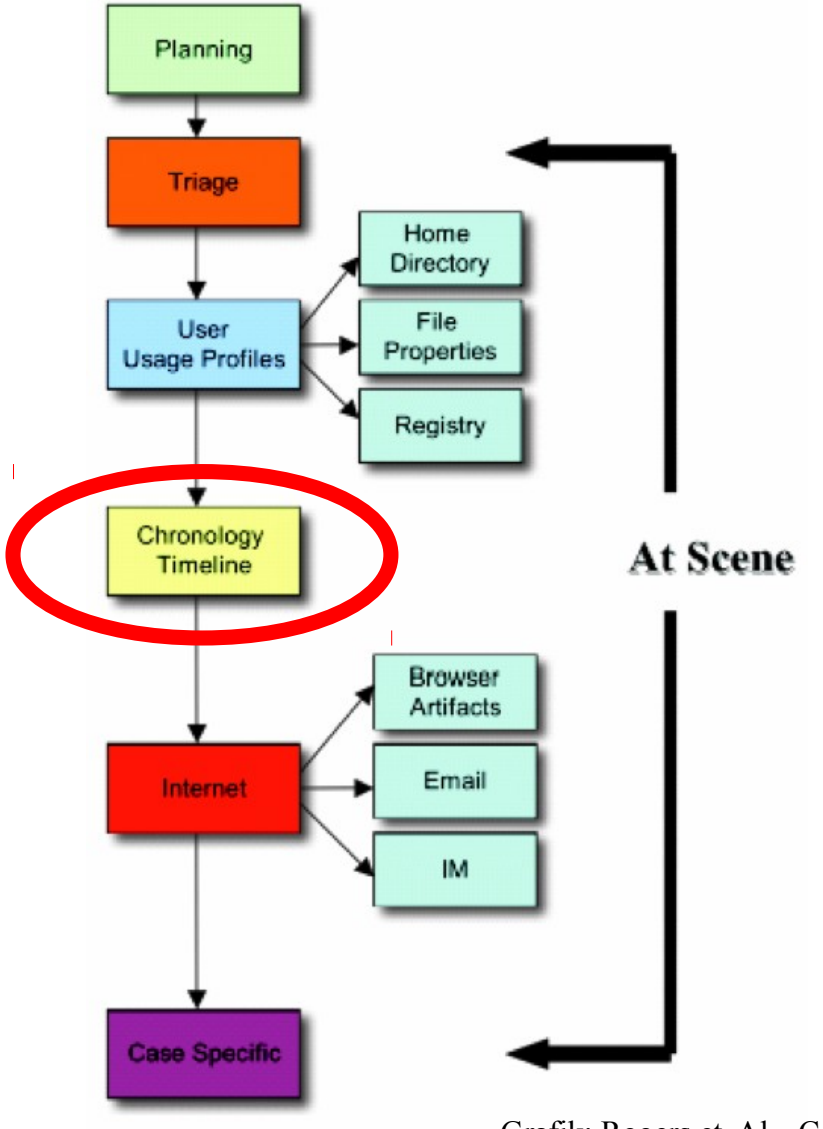
- Ergebnisorientiertes, agiles Vorgehen
- Stabilität, semantische Integrität
- Komplexe Daten, beschränktes Vertrauen





Grafik: Kent et. Al. „Guide to Integrating Forensic Techniques into Incident Response“

Unterstützung durch Timelining-Software

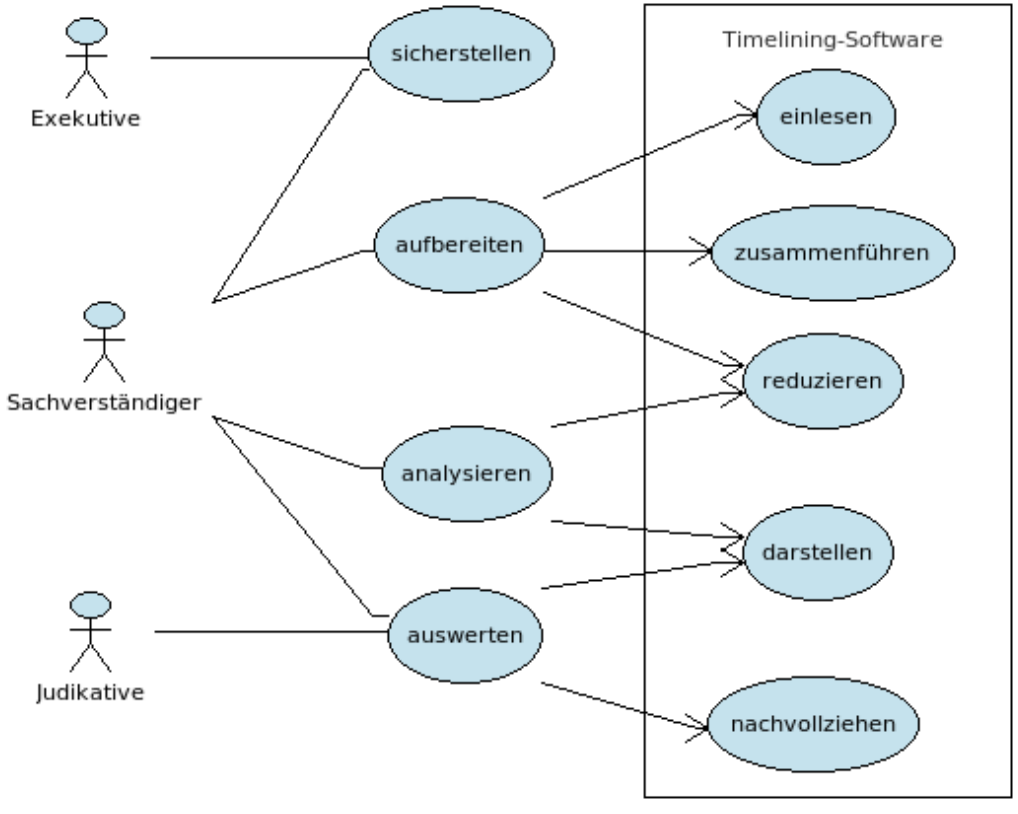


- Auswertung von Zeitdaten
  - Fallen als Metadaten zu Ereignissen an
  - Verbinden unterschiedliche Quellen
  - Kleinster gemeinsamer Nenner
  - Erlauben Rückschlüsse auf Verhalten
- Automatische Visualisierung
  - Ermöglicht optische Auswertung
  - Nutzt die menschliche Intelligenz
  - Vereinfacht die Kommunikation

Grafik: Rogers et. Al. „Computer Fornesics Field Triage Process Model“

Anforderungen an eine Timelining-Software

- Schnellere Resultate als durch forensische Kopien
- Zielgerichtetes Vorgehen
- Standardisierte Dateiformate
  - CSV als Input-Format
  - SVG als Output-Format
- Flexible, iterative Aufbereitung
- Darstellung von Unschärfe
- Identifikation von Beweisen



Sicherstellung von Zeitdaten

Methodik hängt stark vom Analyseobjekt ab:

- Geräteklasse und Art des Speichermediums
- Dateisysteme und Datentypen
- Log- und Metadaten

Einflussfaktoren auf die Aussagekraft:

- Live vs. Post-Mortem Analyse
- Initiierung einer Beweismittelkette
- Uhrenvergleich mit einer Referenzzeit

Sicherstellung von Zeitdaten

- Analyse der Verwendung eines Apple Mail Clients:

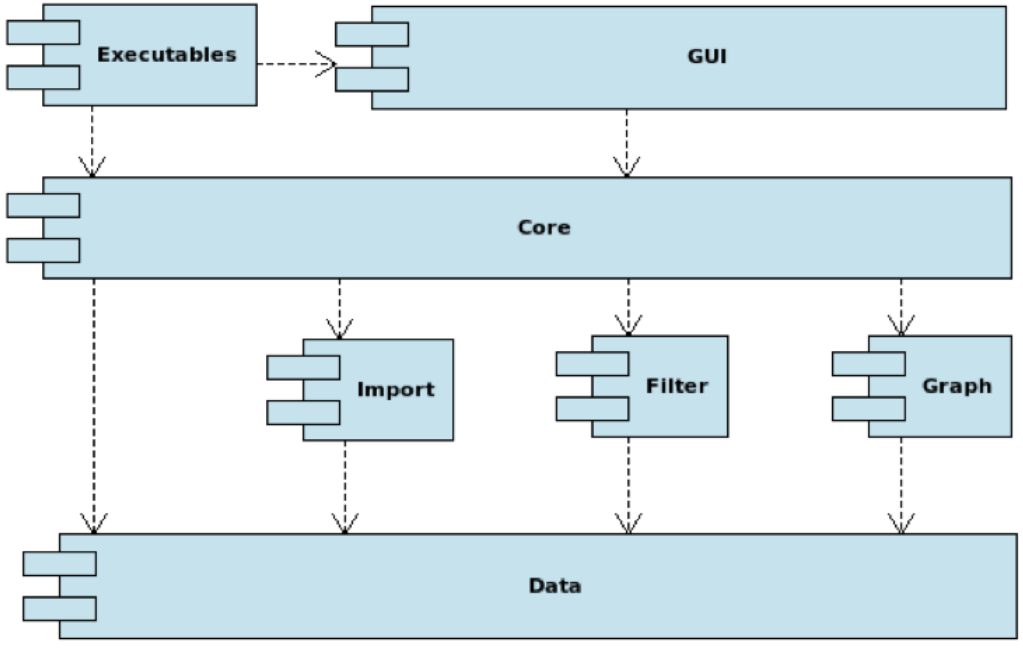
```
$ find Library/Mail -iname \*.emlx -exec ls -liFAt {} \; > mail_modified.txt
$ find Library/Mail -iname \*.emlx -exec ls -liFAtc {} \; > mail_changed.txt
$ find Library/Mail -iname \*.emlx -exec ls -liFAtu {} \; > mail_access.txt
```

- Analyse der Verwendung von Email:

```
$ find Library/Mail/ -iname \*.emlx -exec grep -H "^Date:" {} \; > mail_date.txt
```

- Dateisystem- versus Protokoll-Metadaten

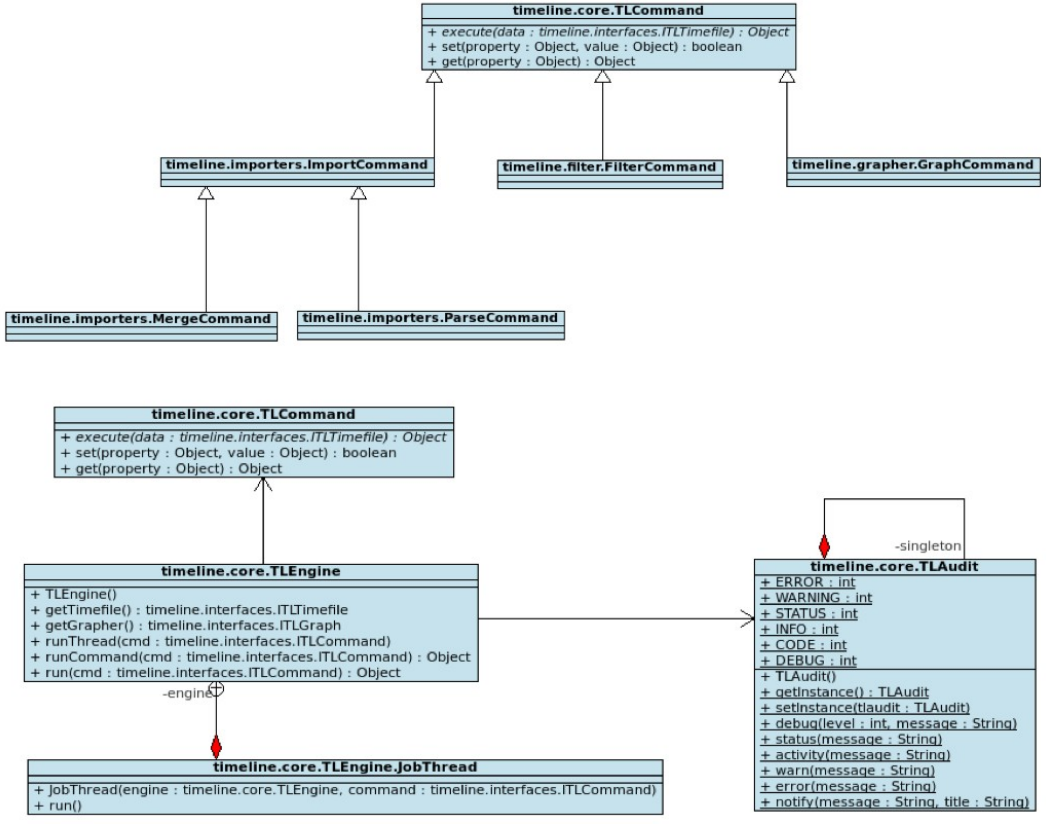
Software-Architektur



- Java SE Plattform
- Desktop-Applikation
- 3 Schichten Architektur
- Model-View-Control
- Datenhaltung im RAM

Software-Architektur

Anwendungskern



Ablauf der Visualisierung

1. Min und Max bestimmen
2. Skalierung festlegen
3. Zeitstrahl zeichnen (SVG)
4. Events einzeichnen (SVG)
5. Audit-Log als Kommentar

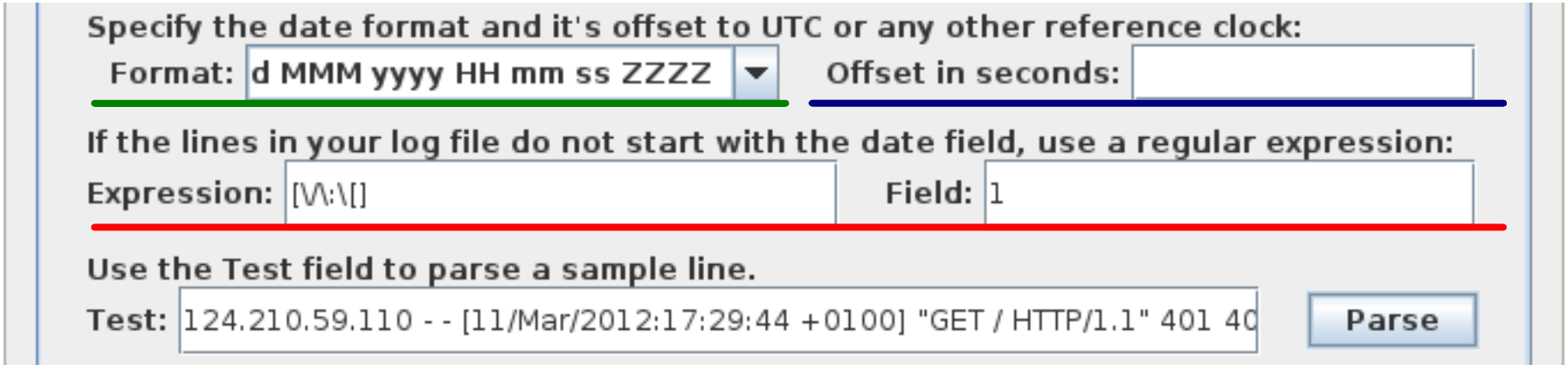
Software-Design

- Datenmodell EVA
- Import- / Merge-Funktion

Eingabedaten: 134.30.19.110 – [11/Mar/2012:17:29:44 +0100] „GET / HTTP/1.1“ 401 409

Datums-Format: d MMM yyyy HH mm ss ZZZZ Abweichung zur Referenzzeit?

Delimiter: [/ : Datum ab Feld Index 1 (bei 0..n)



Ausgabe: 133148334000 2012-03-11 17:29:44 134.30.19.110 – [11/Mar/2012:17: ...

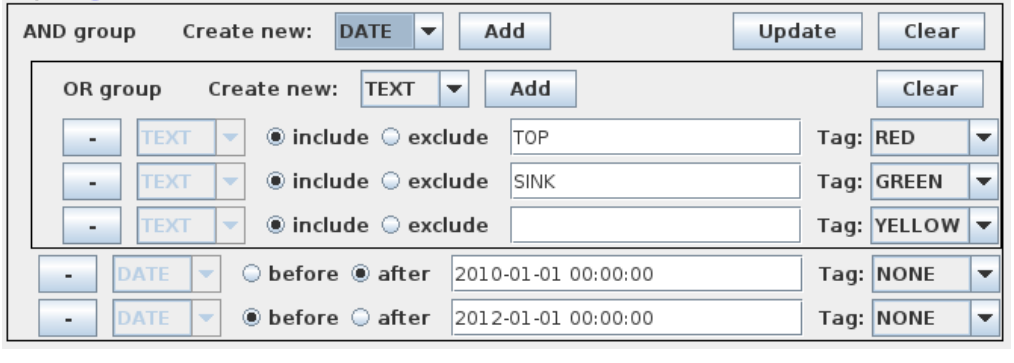
Software-Design

Die Nadel im Heuhaufen:

- Was ist gesucht?
- Was dient als Beweis?
- Wie ist die Aussagekraft?
- Sach- und Fachwissen erforderlich
- Fallspezifisch

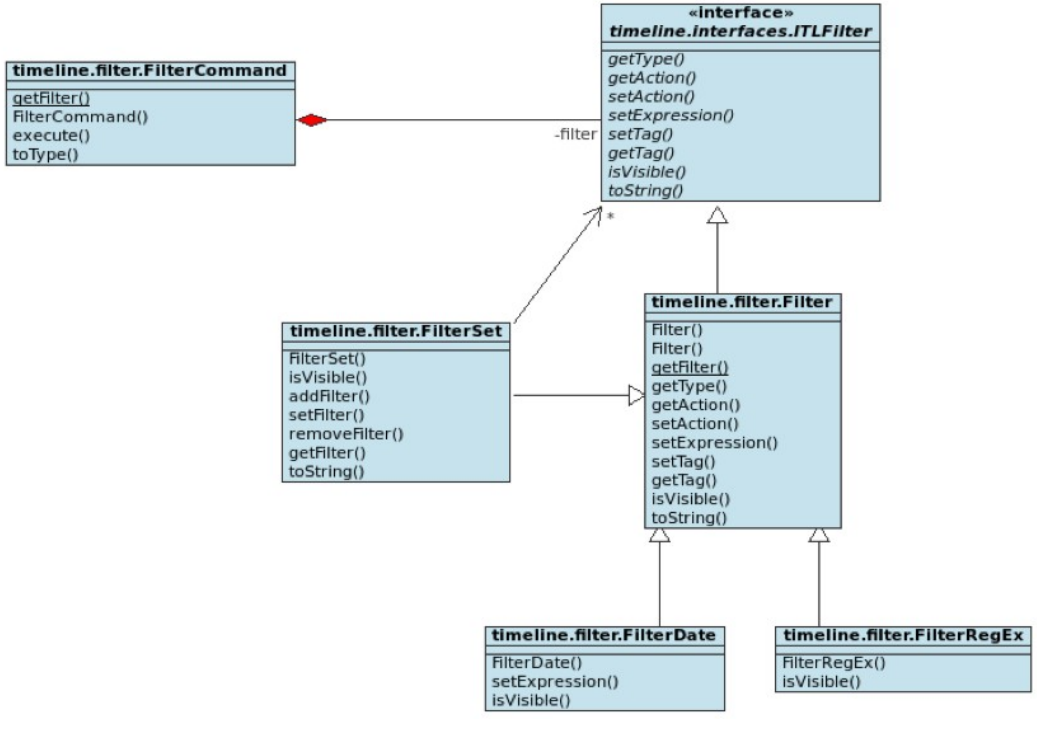
Filter-Typen:

- Textfilter
- Reguläre Ausdrücke
- Zeitliche Filter
- Und- und Oder-Gruppen

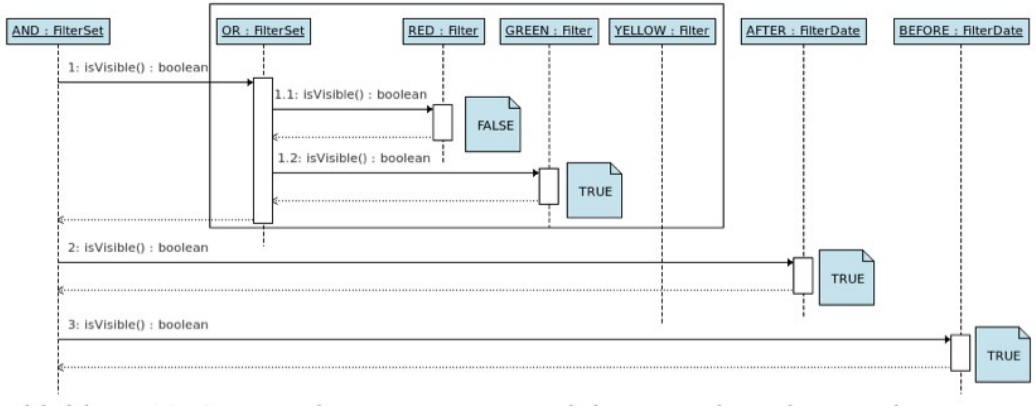


Software-Design

- Filter-Hierarchie



- Filter-Evaluation



Demonstration

- Einlesen und Zusammenführen von Zeitdaten
- Reduktion der Datenmenge
- Visualisierung der Zeitachse
- Analyse mittels Vektorgrafiksoftware

Schlussbetrachtung

- Iteratives Vorgehen für bessere Resultate
- Fokus auf fallrelevante Fragenstellung
- Flexible SW-Architektur
- Aufbauend auf Standards
- Einfaches aber mächtiges Analysewerkzeug

Performance:

- Schnelle Generierung
- Limitierte Darstellung

Nutzen für die Praxis:

- Visualisierung
- Automatisierung
- Standardisierung

Danke für die Aufmerksamkeit