PostGIS Versionsverwaltung

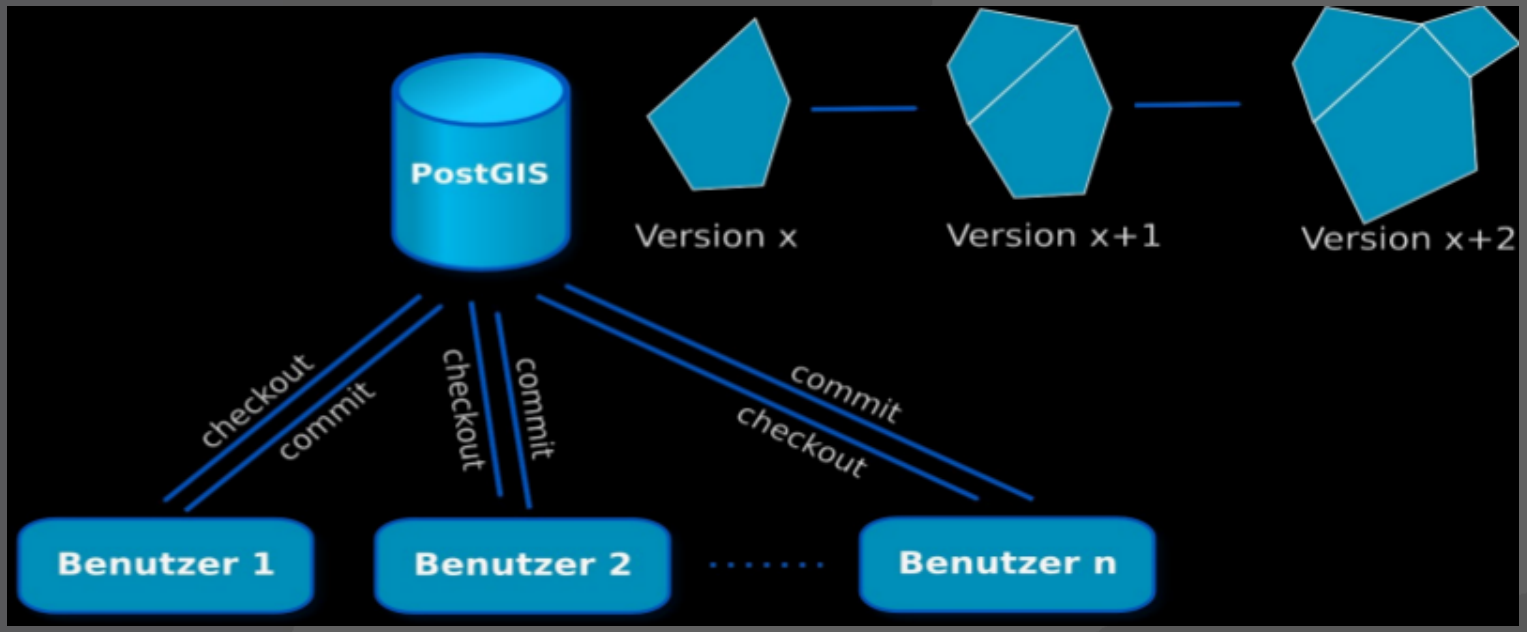
Dr. Horst Düster
Sourcepole AG, Zürich
Twitter: @sourcepole
www.sourcepole.com

Versionsverwaltung

- **System zum Erfassen von Änderungen an Daten**
  - Protokollierungen der Änderungen
  - Wiederherstellung alter Stände
  - Archivieren der einzelnen Stände eines Datums
  - Koordinieren des gemeinsamen Zugriffs
  - Gleichzeitige Entwicklung mehrerer Entwicklungszweige (engl. Branches)

Versionsverwaltung mit pgvs

- **Versionsverwaltung für PostGIS (pgvs)**
  - Gleichzeitiges Editieren eines PostGIS Layers durch mehrere Benutzer
  - Konfliktlösung konkurrierender Objekte
  - Unterschiedliche Versionsstände der einzelnen User
  - Archiv über Raum und Zeit (Historisierung)



Versionsverwaltung mit pgvs

- **Dokumentation**
  - https://github.com/sourcepole/pgversion
- **pgvs**
  - Logik der Versionierung in der DB
- **QGIS Plugin**
  - Anbindung an pgvs aus QGIS

Versionsverwaltung mit pgvs

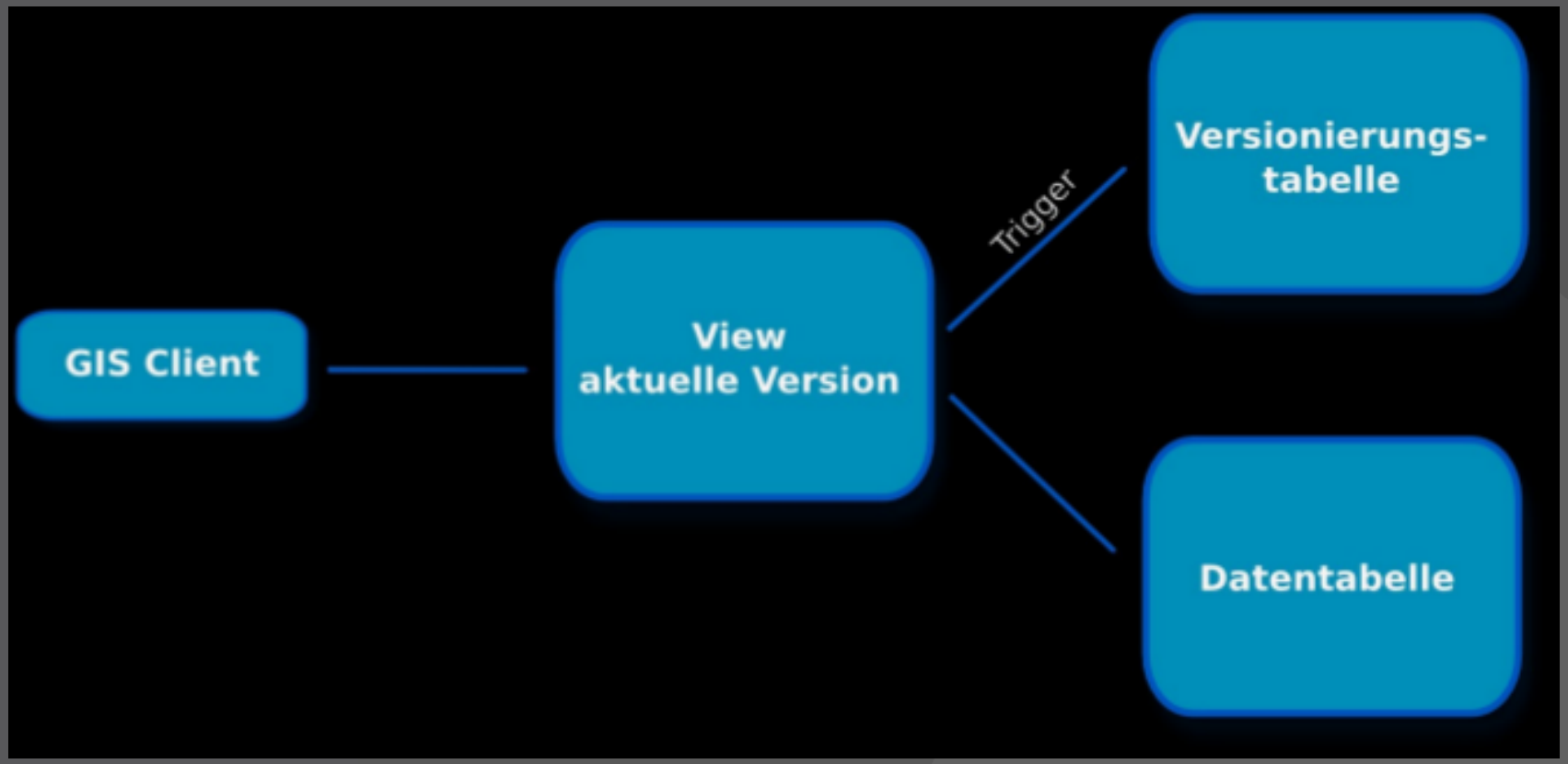
pgvsinit()	Initialisieren der pgvs Umgebung
pgvscommit()	Einpflegen ins Repository
pgvsmerge()	Konfliktlösung
pgvsrevert()	Zurücksetzen auf HEAD Revision
pgvslogview()	Anzeigen der Log-History
pgvsrollback()	Die HEAD Revision durch frühere Revision ersetzten
pgvscheckout()	Auszug einer bestimmten Version

Versionsverwaltung mit pgvs

- **Vorbereitung**
  - pgvs in der DB einrichten
  - Layer mit der Prozedur versions.pgvsinit(<Tabelle>) in das Versionierungssystem aufnehmen

=> neuer DB-View ist angelegt, über den zukünftig gearbeitet wird

Versionsverwaltung mit pgvs



Versionsverwaltung mit pgvs

- Änderungen übertragen

```
select *
from versions.pgvscommit('schema>.<tablename>',
                         'Log message')
```

- Ergebnis:

  Liste der Objekte, die in Konflikt zu Änderungen anderer User stehen

Versionsverwaltung mit pgvs

- **Konflikte lösen**

*select * from versions.pgvsmerge('<schema>.<table>,*
<record-id>,
'<username>')

- **Ergebnis:**

  Das Objekt mit der entsprechenden ID wird durch das Objekt des Users ersetzt.

Versionsverwaltung mit pgvs

- Änderungen zurücksetzen

  *select * from versions.pgvsrevert('<schema>.<table>')*

- Ergebnis:

  Setzt alle Änderungen des aktuellen Users zurück auf die HEAD Revision der Versionierung

Versionsverwaltung mit pgvs

- **Auf eine bestimmte Version zurücksetzen**

*select * from versions.pgvsrollback('<schema>.<table>', Version)*

- **Ergebnis:**

  Ersetzt die aktuelle HEAD Version durch die gewünschte frühere Version.

Versionsverwaltung mit pgvs

- Logs ansehen

  *select * from versions.pgvslogview('<schema>.<table>')*

- Ergebnis:

  Auflisten aller Commits mit den zugehörigen Logeinträgen

Versionsverwaltung mit pgvs

- **Version auschecken**

*select * from versions.pgvscheckout('<schema>.<table>', Versionsnummer)*

- **Ergebnis:**

  Auszug der Records der gewünschten Version

Versionsverwaltung mit pgvs

- **Demo pgvs mit QGIS-Plugin**
  - Neues QGIS Projekt
  - Versionierten Layer für Benutzer hinzufügen
  - Layer bearbeiten
  - Änderungen übernehmen
  - Verwalten der Versionen über den LogView
  - Checkout einer bestimmten Revision
  - Rollback einer Revision zur aktuellen Version

Versionsverwaltung mit pgvs

- **Ausblick**
  - Zweige verwalten (Branches)
  - Zweige verbinden (Branches Merge)
  - Verbesserter Differenzenmonitor

Haben Sie Fragen?

Dr. Horst Düster
horst.duester @ s o u r c e p o l e . c h