

CAS FHNW Spatial Data Analytics

Infrastruktur der Datenanalyse



Lektion	Zeit	Inhalt
1	08:45 – 09:30	Einführung ins Thema <i>Was ist Infrastruktur? Spatial Data Infrastructure (SDI)</i>
2	09:30 – 10:15	Computational Notebooks <i>Python und Jupyter</i>
3	10:30 – 11:15	Daten, Dateien, Dienste / Datenbanken <i>gängige Formate, PostgreSQL und PostGIS, Geodatabase</i>
4	11:15 – 12:00	Standards <i>EPSG, OGC, Simple Features, WMS etc., Basis-Standards</i>
5	13:00 – 13:45	Isolation und Automation <i>Environments, Container, Skripte</i>
6	13:45 – 15:30	Aspekte aus der Produktion <i>Experteninterview</i>
7	14:45 – 15:30	ArcGIS als Plattform für Analysen <i>Demonstration durch einen Praktiker</i>
8	15:30 – 16:15	Cloud und Review <i>service models, deployment models, Binder Project, ArcGIS</i>

Didaktisches

Fragen stellen

- jederzeit

Notizen machen

- Skript ergänzen (Slides sind meine Stütze und werden nicht verteilt)
- Probleme sammeln (→ lessons learned)
- Glossar führen

Demos nachvollziehen

- Probleme werden auftauchen
- Wir werden vermutlich nicht die Zeit haben, alle zu lösen

Personelles

Urs-Jakob Rüetschi, Dr. sc. nat. UZH, Dipl. geogr., Esri Suisse, Dira GeoSystems

- steht vor Ihnen

Emanuel Mahler, CEO, Dira GeoSystems

- Experteninterview zu Aspekten der Produktion (Nachmittag)

Stefan Graf, Esri Suisse

- Betrachtungen zu ArcGIS als Plattform für die Datenanalyse (Nachmittag)

Infrastruktur

...ein schwieriges Thema gezogen...

Infrastruktur

Architektur

Plattformen

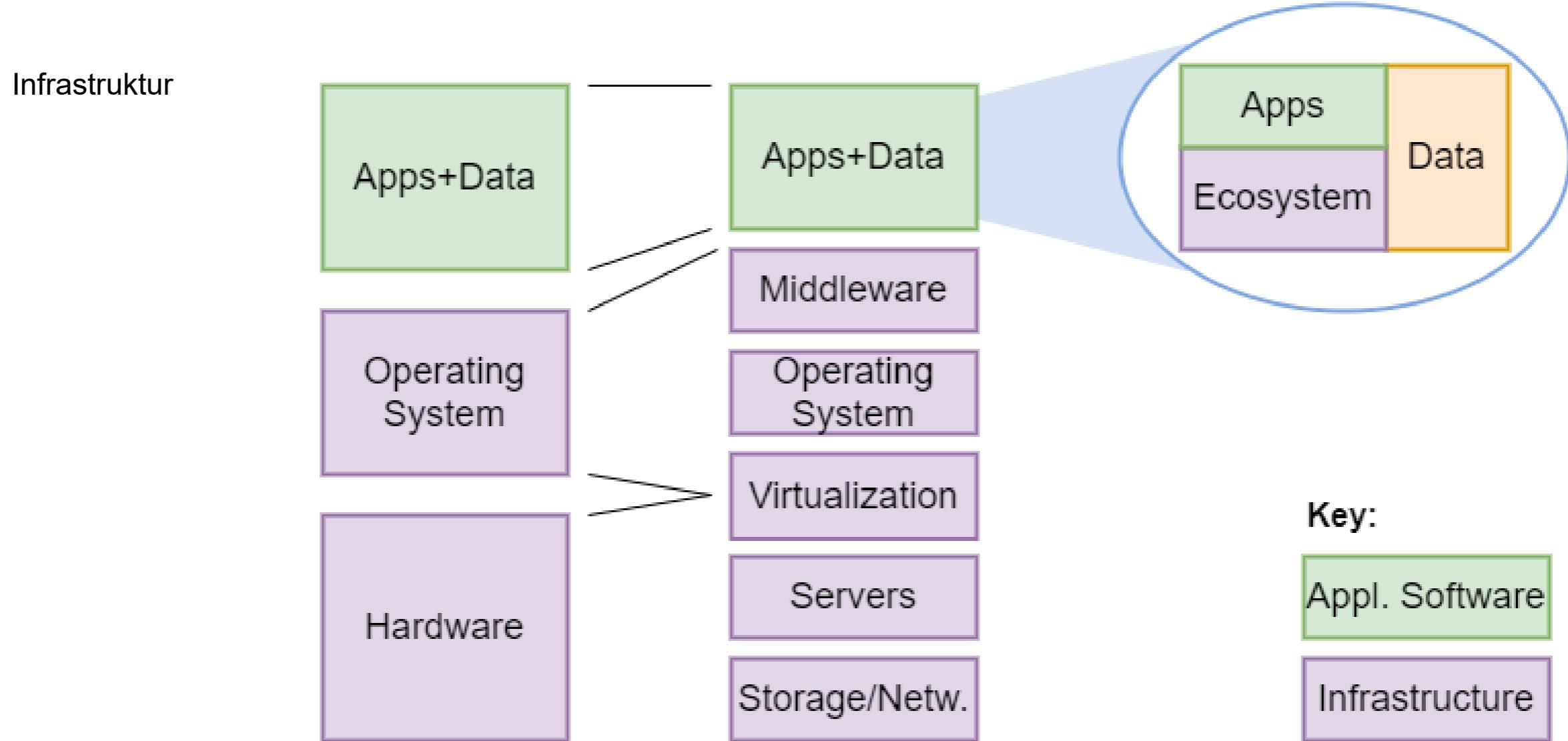
Systeme

«Ökosysteme»

Stacks

Hardware

Environments



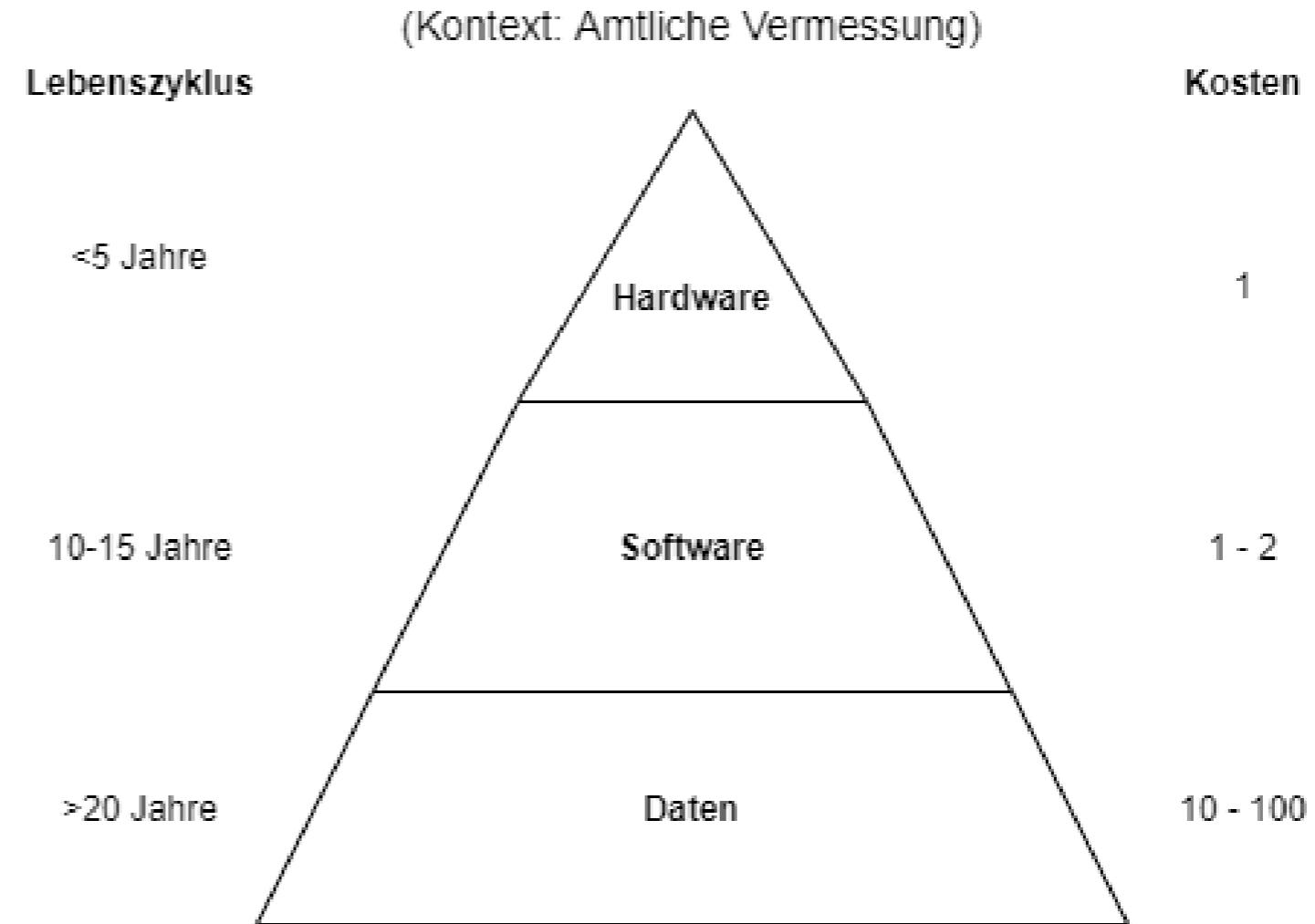
Layers in a system for data analysis (and in computer systems in general) (source: the author)

Betriebssysteme



Infrastruktur

Lebenszyklus und Kosten



Quelle: aus einem GIS-Skript der Uni Zürich von ca. 1998

Praxis

Systemanforderungen für Esri ArcGIS Pro?

Installation Windows Terminal

Installation Windows Subsystem for Linux

Skizze: folgende Begriffe in Beziehung setzen:

- Infrastruktur • Plattform • Stack • Ökosystem • System • Anwendung

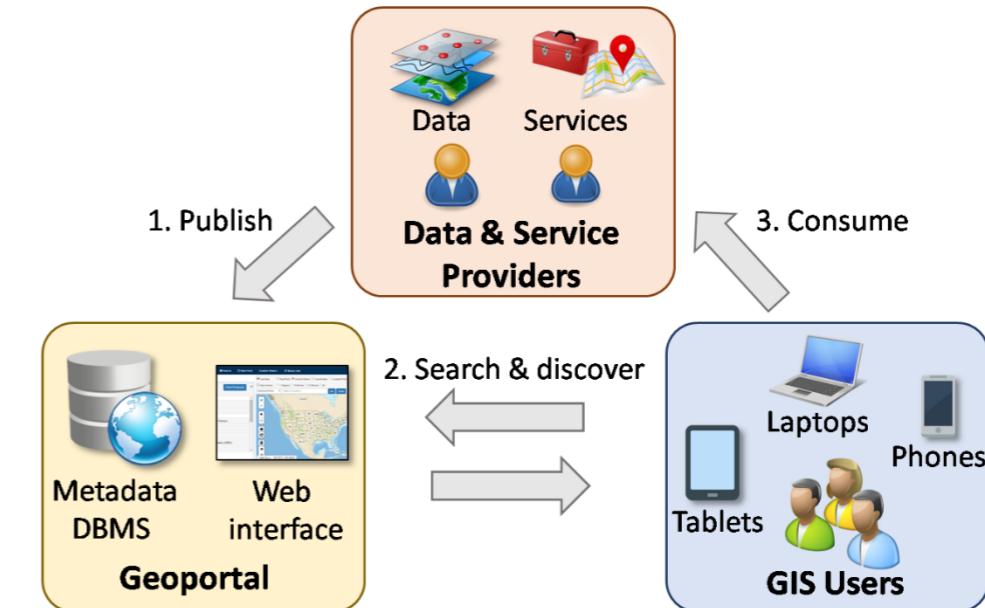
Spatial Data Infrastructure

Spatial Data Infrastructure (SDI)

Ziel: räumliche Daten finden und beziehen

Die wichtigsten Komponenten

1. Geoportal (user interface)
2. Metadaten (suchbar)
3. Suchfunktion (fachlich und räumlich)



The publish-find-bind pattern

Literatur: Y. Hu, W. Li, 2017, “Spatial Data Infrastructures,” in *The Geographic Information Science & Technology Body of Knowledge* (2nd Quarter 2017 Edition), John P. Wilson (ed.), DOI [10.22224/gistbok/2017.2.1](https://doi.org/10.22224/gistbok/2017.2.1)

Spatial Data Infrastructure (SDI)

- geo.admin.ch – das Geoportal des Bundes
- geocat.ch – Katalog der Geodaten der Schweiz (Bund, Kantone, Gemeinden, Private)
- opendata.swiss – Portal für Open Government Data schweizweit (keine *Spatial DI*, aber *auch* räumliche Daten)
- geolion.zh.ch – Geodatenkatalog Kanton Zürich
- www.stadt-zuerich.ch/geodaten – Geodaten der Stadt Zürich
- [INSPIRE](#) – INfrastructure for SPatial InfoRmation in Europe. Geodateninfrastruktur in der EU, eine Richtlinie, welche die Mitgliedsstaaten zur Bereitstellung von Geobasis- und -fachdaten verpflichtet (über Netzdienste).
- data.gov – U.S. Government Open Data
- en.wikipedia.org/wiki/List_of_GIS_data_sources – keine SDI, aber eine nützliche Sammlung

Spatial Data Infrastructure (SDI)

Erste SDI in USA 1993, die «National Spatial Data Infrastructure»

Name rückt Daten in die Nähe von ‘klassischer’ Infrastruktur à la Strassennetz und Stromleitungen

Schweiz: Bundes Geodaten-Infrastruktur (BGDI) basiert auf GeolG von 2008; darin:

Dieses Gesetz bezweckt, dass Geodaten über das Gebiet der Schweizerischen Eidgenossenschaft den Behörden von Bund, Kantonen und Gemeinden sowie der Wirtschaft, der Gesellschaft und der Wissenschaft für eine breite Nutzung, nachhaltig, aktuell, rasch, einfach, in der erforderlichen Qualität und zu angemessenen Kosten zur Verfügung stehen. (Art. 1 GeolG)

www.geo.admin.ch

[Bundes Geodaten-Infrastruktur \(admin.ch\)](http://Bundes Geodaten-Infrastruktur (admin.ch))

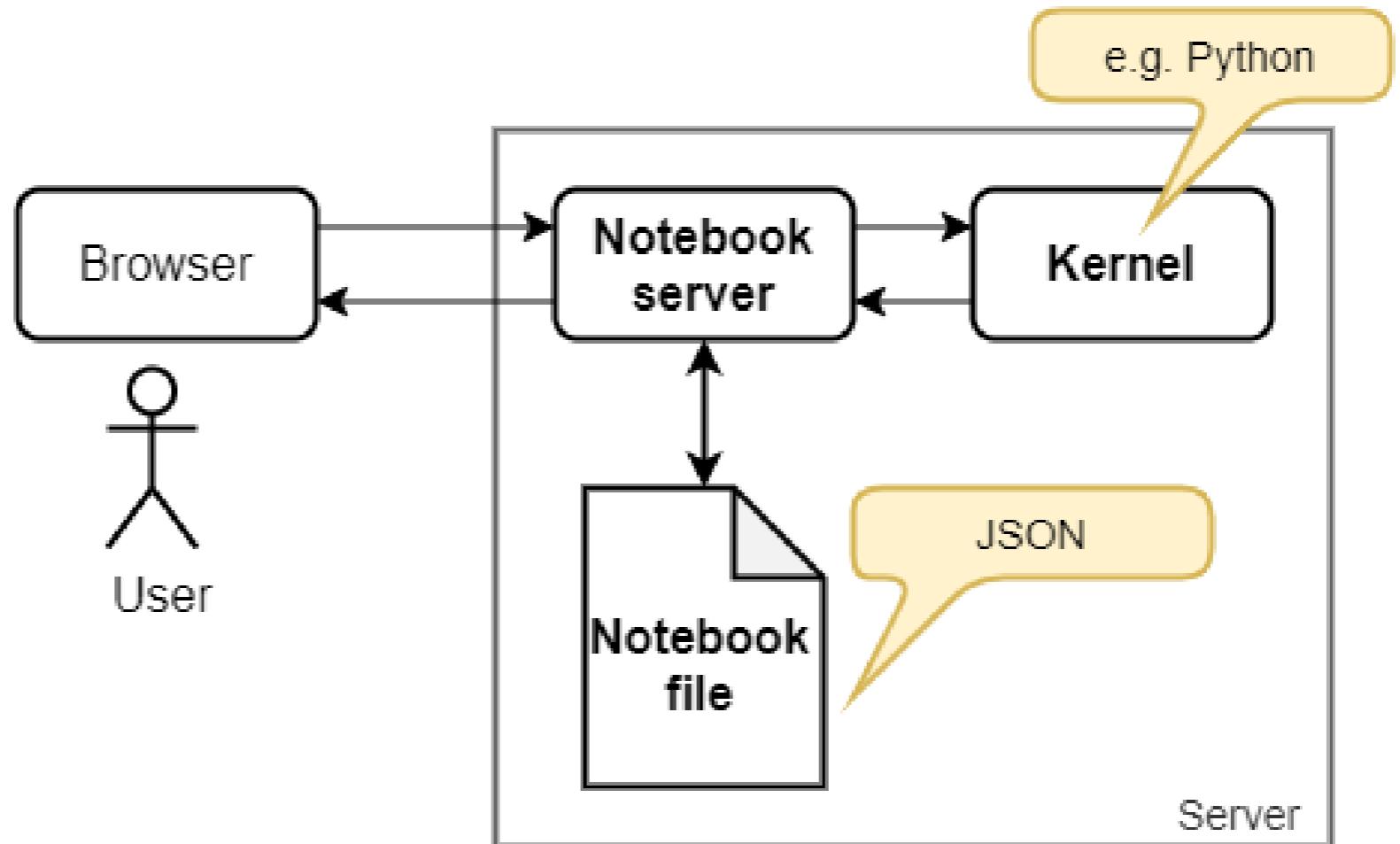
Computational Notebooks

Computational Notebooks

Paradigma

der Interaktion mit einem
Computer

Notebook Architecture



Source: author's drawing after the similar illustration on jupyter.org

Computational Notebooks — Praxis

Anaconda installieren ([Übung](#))

- bequeme Python-Distribution: alles dabei

Platzhirsch im Bereich Data Science: Jupyter

- gründlich ausprobieren ([geopandas](#))

Alternativen: Mathematica – hat Notebooks seit gut 30 Jahren

- kurze Demo ([Mathics](#), evtl. Filmchen zu Geo mit Mathematica)

Alternativen: Observable – Fokus Kollaboration

- [observablehq.com](#) – D3.js ([d3js.org](#))

Python-Paket-Manager conda vs. pip

- evtl. Installation in Ubuntu (WSL) als [Demo/Übung](#)

Computational Notebooks

Inwiefern sind Notebooks ein neues Paradigma in der Nutzerinteraktion mit Computern?

Von «point-and-click» Desktop-Anwendungen zu Code basierten Notebooks: Vor- und Nachteile?

Zustimmung? Warum/nicht?

Notebooks sind einfach mit anderen zu teilen.

Notebooks begünstigen eine offene Wissenschaft (open science).

Notebooks sind selbst-dokumentierend.

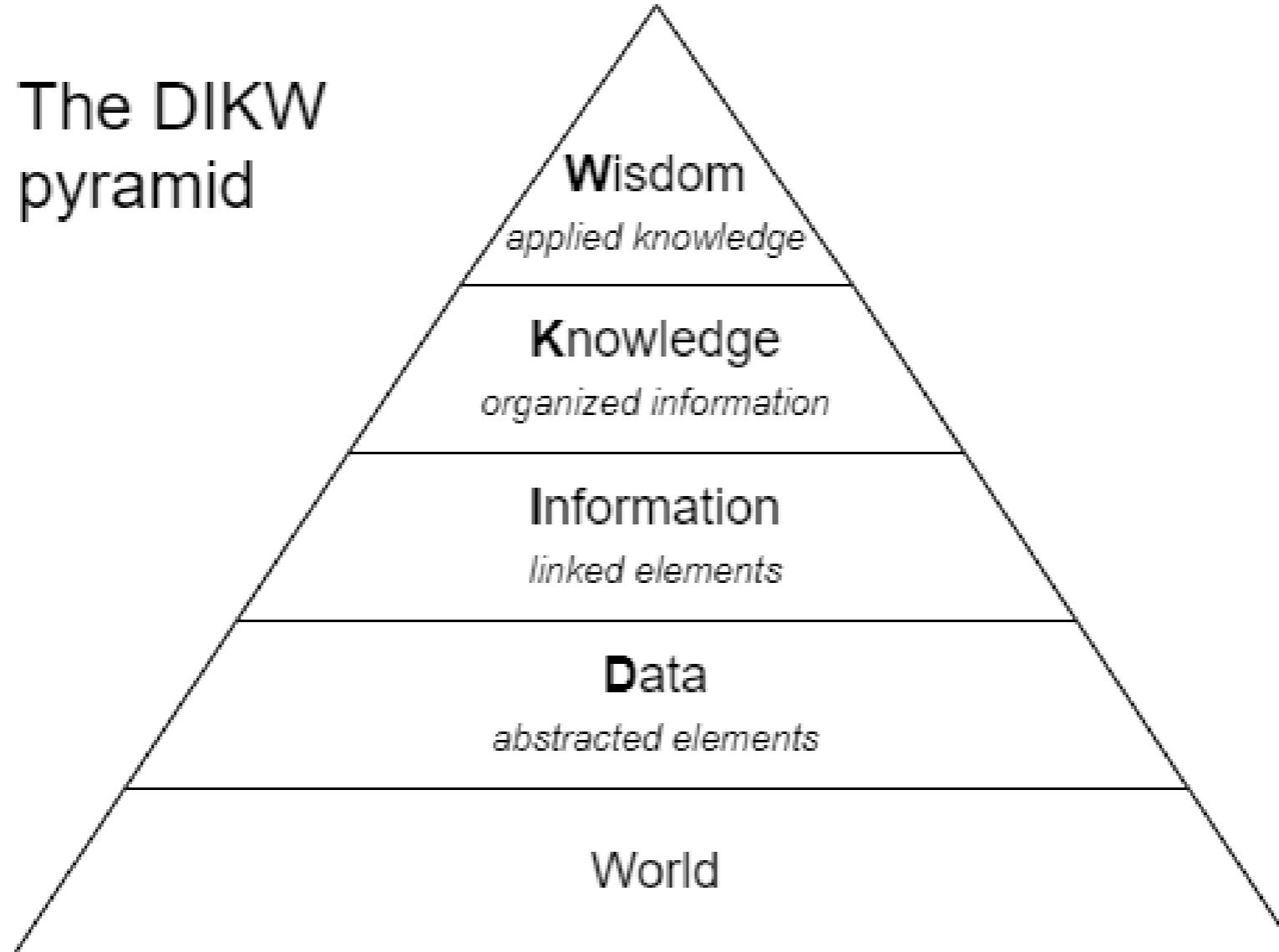
Notebooks sind wiederholbar/reproduzierbar.

Woher kommen die importierten Python-Module?

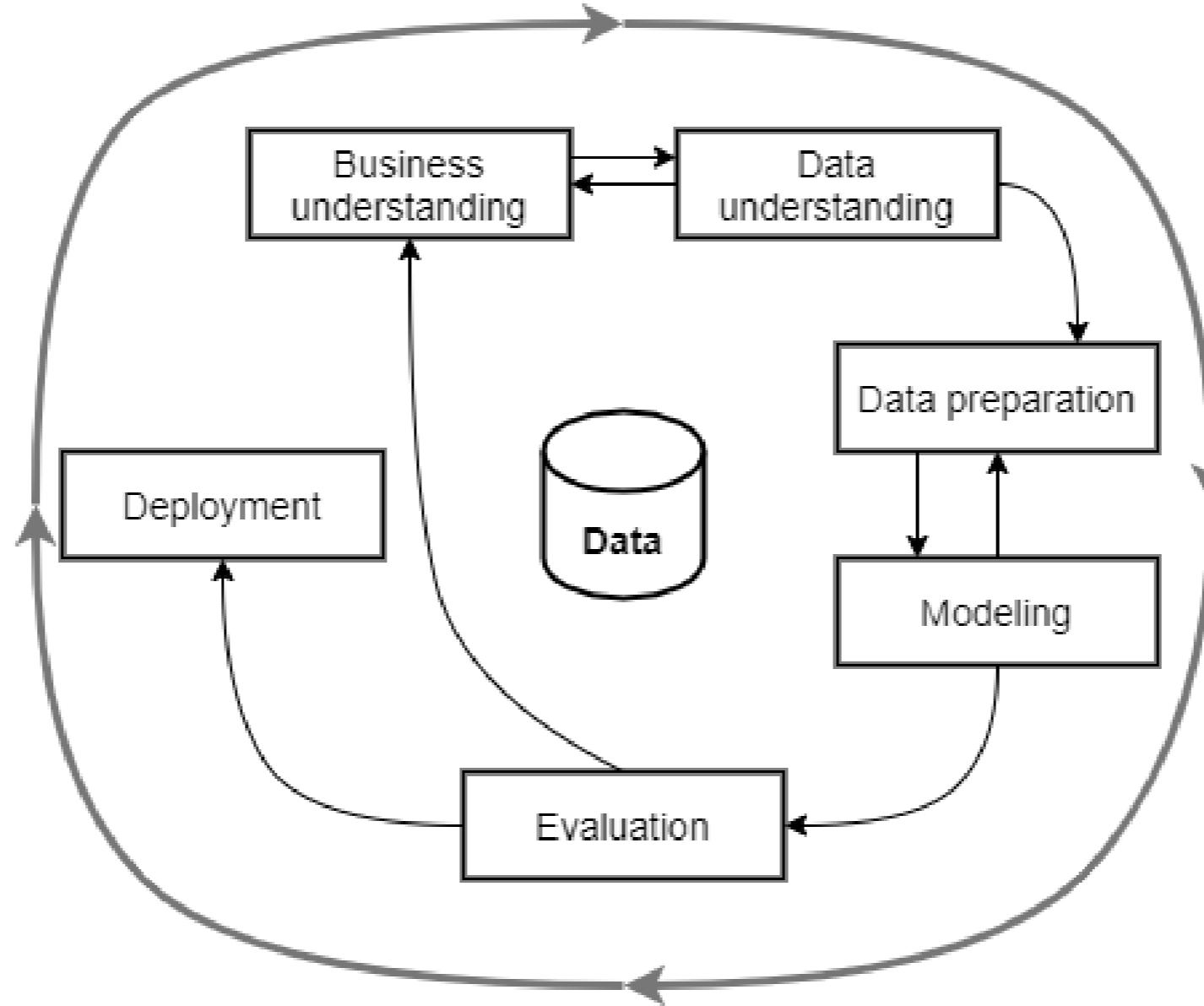
Eignen sich Notebooks für den Übergang in die Produktion?

Daten, Dateien, Dienste

The DIKW pyramid

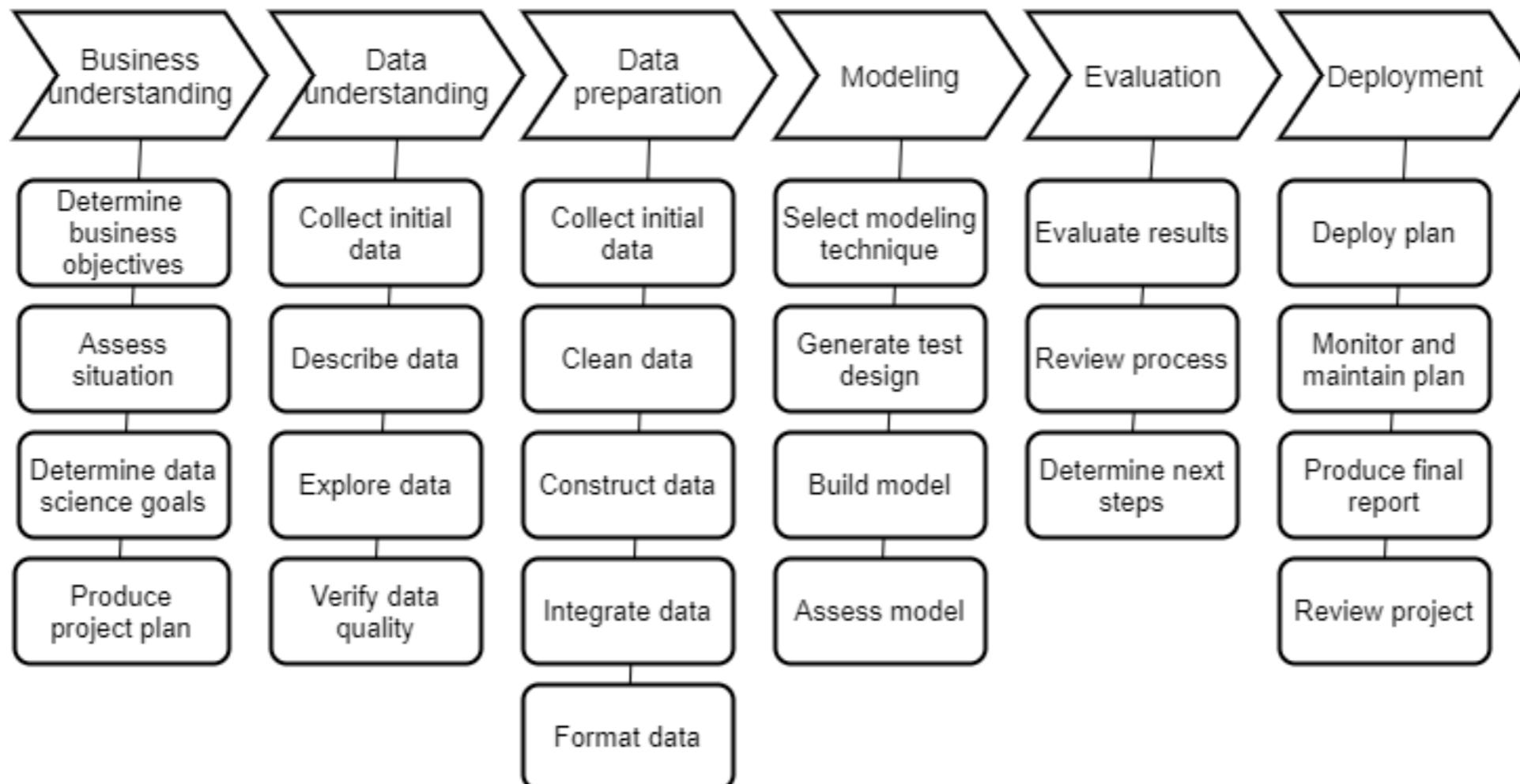


Quelle: Kelleher und Tierney, *Data Science*, MIT Press 2018, p.55



Quelle: Chapman et al., CRISP-DM 1.0: Step-by-step data mining guide, 2000, p.13.

Stages and tasks in the CRISP-DM



Quelle: Chapman et al., CRISP-DM 1.0: Step-by-step data mining guide, 2000, p.15, simplified.

Perspektiven auf Daten

strukturiert / unstrukturiert

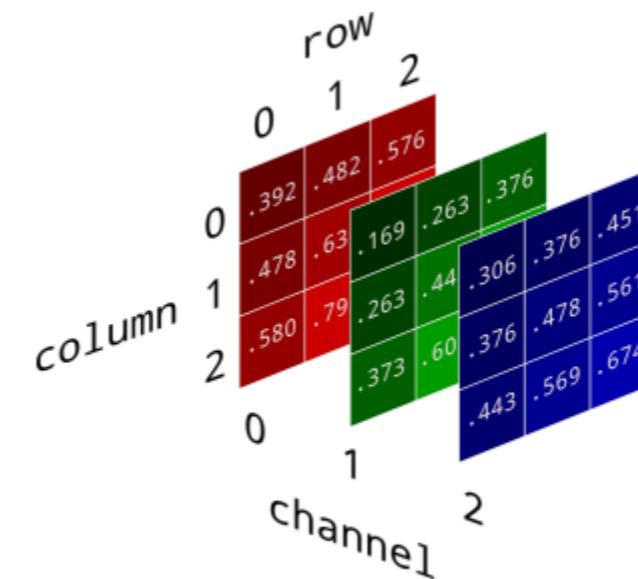
Rohdaten / abgeleitet

bewusst erfasst / beiläufig angefallen

VGI

Geographische Daten

Abstraktion	gängige Repräsentation	Beispiele
Objekte	Features	Admin. Grenzen
Felder	Raster, TIN	Niederschlagsmenge
Netzwerke	Graphen	Verkehrsnets



Quelle: Slides von R. Leiterer, 2022

Häufige Dateiformate in der räumlichen Datenanalyse ([Input](#))

Plain Text (.txt, .dat) – Encoding, Spaltenformat oder Separatoren, Multiline Records, AWK

CSV (.csv) und **TSV** (.tsv) – Varianten, nicht standardisiert, Library verwenden (Python: [csv](#))

Excel (.xls/.xlsx) – sehr populär, auch ausserhalb Microsoft gut unterstützt

Shapefile (.shp) – de facto Standard für Vektordaten für GIS, [Esri White Paper](#)

GeoJSON (.geojson, .json) – Vektordaten mit Attributen in JSON

GeoPackage (.gPKG) – Vektor und Rasterdaten in einer SQLite-Datenbank, OGC Standard

GML (.gml) – ein XML-Format, erweiterbar, OGC Standard

GeoTIFF (.tif, .tiff) – oder reines TIFF plus [World File](#) (.tfw)

NetCDF (.nc4) – für mehrdimensionale Rasterdaten, für wissenschaftliche Daten

Häufige Datenformate für die Präsentation ([Output](#))

PNG (.png) – Rasterformat mit verlustfreier Kompression, versch. Modi und Tiefen

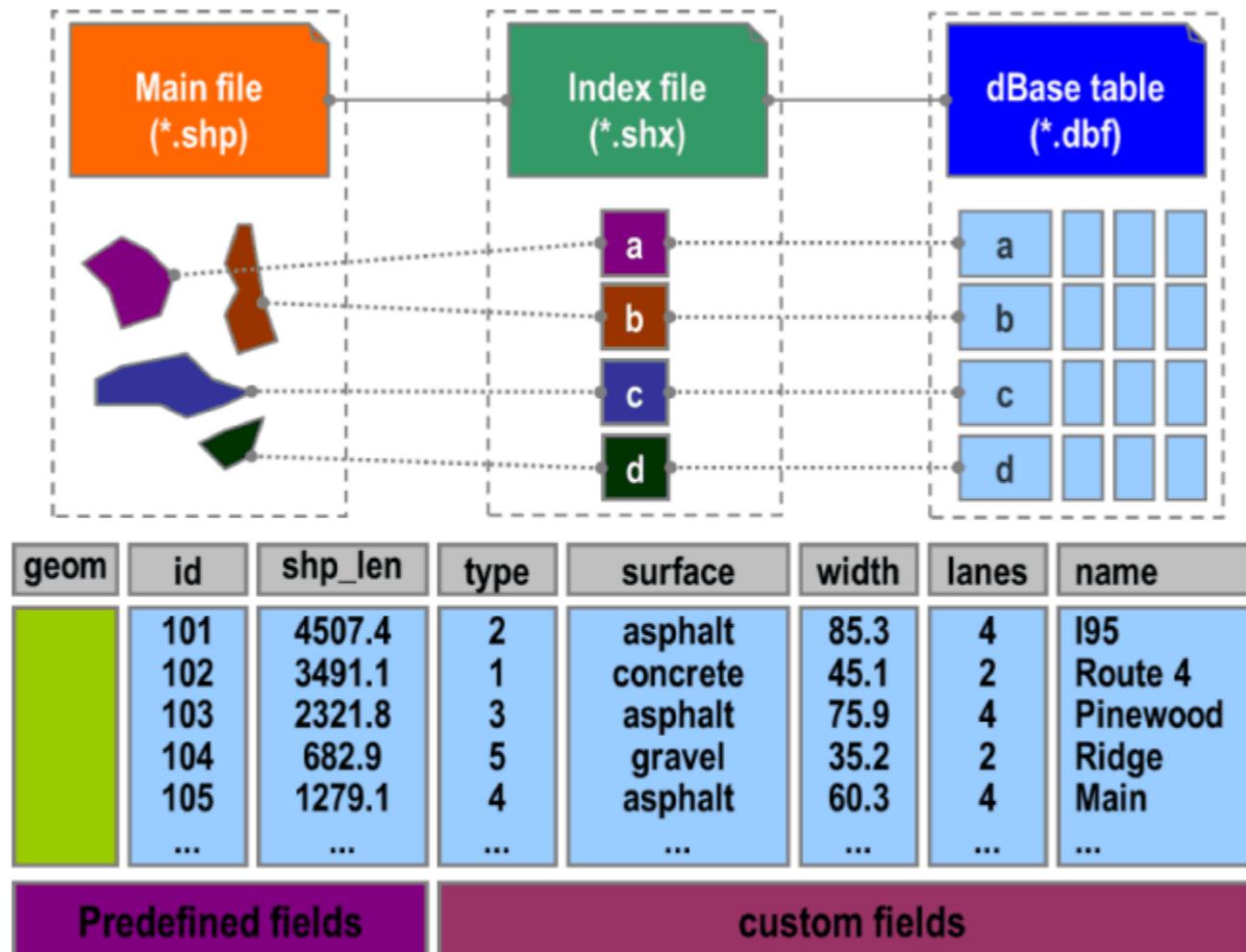
JPEG (.jpg, .jpeg) – Rasterformat mit verlustbehafteter Kompression

PDF (.pdf) – einzelne Grafiken bis ganze Bücher, immer breiter unterstützt

EPS (.eps) – Vektorgrafiken, vor Verbreitung von PDF beliebt

SVG (.svg) – Vektorgrafik in XML, kann direkt im Web Browser angezeigt werden

Markdown (.md) – einfacher geht nicht – immer breiter unterstützt – jeder Projektordner braucht ein top-level README.md 😊



Quelle: Slides von R. Leiterer, 2022

Daten via APIs

Beispiel 1: opendata.swiss (siehe Kapitel zu SDI; hat API zur Suche)

Beispiel 2: [Esri's Python API](https://developers.arcgis.com/python/) (Interface zu Portal for ArcGIS, u.a. Suche nach Inhalten)

Räumliche Datenbanken

Datenbanken

Relationales Modell

DBMS, RDBMS

SQL

DDL

DML

CRUD

ACID

NoSQL

In the wild... db-engines.com/de/ranking

```
SELECT <was>
FROM <woher>
WHERE <Bedingungen>
```

Nicht thematisiert:
Zugriffsschutz (permissions, grants)
Konsistenzsicherung (integrity, checks)
Transaktionen (begin, commit, rollback)
Verteilung (CAP theorem*)
Datenbankentwurf und ERM**

* <https://www.ibm.com/cloud/learn/cap-theorem>

** entity-relationship-model

Datenmodell

TODO

Räumliche Datenbank

Erweiterung einer relationalen Datenbank um einen «Spatial Type»

- Räumliche Datentypen (Punkte, Linien, Polygone)
- Funktionen auf räumlichen Datentypen (Konstruktion, Abfrage, Relation)
- Räumliche Indizierung (Performance)

Beispiel 1: **PostGIS** erweitert **PostgreSQL** zur räumlichen Datenbank ([Übung/Demo](#))

Beispiel 2: Esri's **Geodatabase** für diverse RDBMS (und File Geodatabase) ([Demo](#))

Zugriff mit Python/Jupyter auf PostGIS ([Übung/Demo](#))

Processing



Processing

Unsere Definition: Automatisierte Verarbeitung von Daten (batch)

v.a. im Kontext der Datenvorbereitung (data preparation in CRISP-DM)

was geschieht da typischerweise? ([Diskussion](#))

einzelne Schritte, verknüpfbar (Kette)

Frameworks: Esri: [Geoprocessing](#) (GP), QGIS: [Processing](#)

ETL (extract-transform-load) als Spezialfall

FME als Platzhirsch bei Spatial ETL (v.a. viele Formate)

eigene Tools (Prozessschritte) erstellen; attraktiv weil ([warum?](#))

Standards



Standards

Standards helfen beim Austausch von Daten und bei der Nutzung von Diensten:

- Daten – kompatibel
- Prozesse – interoperabel

«Das schöne an Standards ist, dass es so viele zur Auswahl gibt.»

- im Bereich räumlicher Daten nur wenige – echter Mehrwert

Standards

Die Organisationen «hinter» den Standards:

- Open Geospatial Consortium (OGC) – GIS Standards www.ogc.org
- World Wide Web Consortium (W3C) – Web Standards www.w3.org
- Internet Engineering Task Force (IETF) – Internet Standards www.ietf.org
- Marktmacht – de facto Standards

Die drei Standards mit der vermutlich grössten praktischen Bedeutung:

- EPSG, Simple Features, SQL

EPSG Geodetic Parameter Dataset

epsg.org – offizielle Web-Präsenz des EPSG Geodetic Parameter Datasets

epsg.io – direkt Infos zu einer SRID, z.B. epsg.io/4326

spatialreference.org – Informationen zu räumlichen Referenzsystemen

SRID	Bezeichnung	Anmerkungen
4326	WGS 84	World Geodetic System 1984: geographische Koordinaten, GPS
21781	CH1903 / LV03	Schweizerische Landesvermessung 1903: die traditionellen projizierten Koordinaten der Schweiz (Bern bei 600'000 / 200'000)
2056	CH1903+ / LV95	Schweizerische Landesvermessung 1995: die neuen projizierten Koordinaten der Schweiz (Bern bei 2'600'000 / 1'200'000)
3857	WGS 84 / Pseudo-Mercator	de facto standard for Web Maps, Google maps since 2005; von echten Geodäten nicht anerkannt, weil sphärische Behandlung von ellipsoidischen (WGS84) Koordinaten

OGC Simple Features

www.opengeospatial.org/standards/sfa

www.opengeospatial.org/standards/sfs

Objektmodell

Geometriemodell

WKT und WKB (Serialisierung)

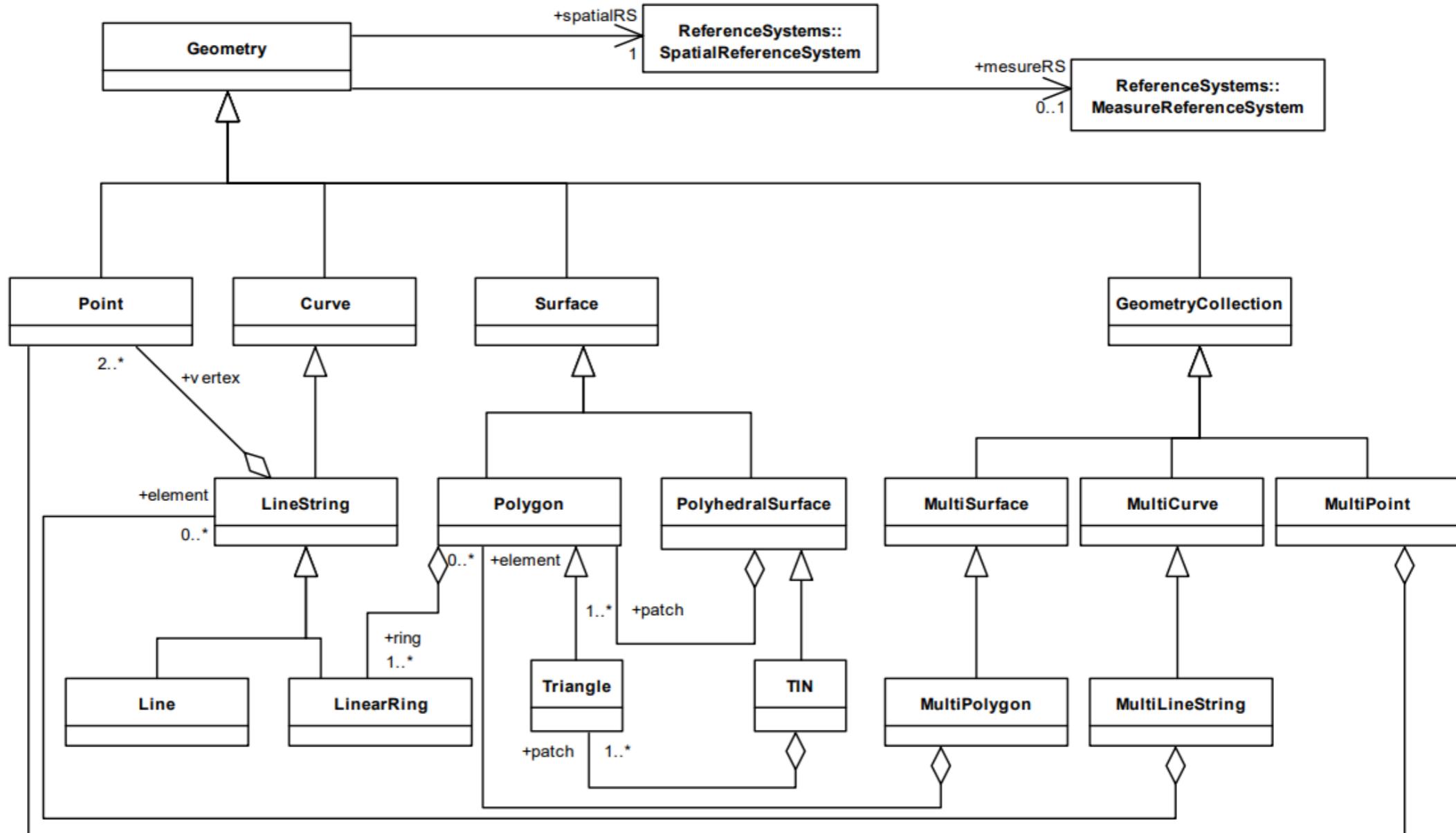
Metatabellen

A: LINESTRING (0 0, 2 0)

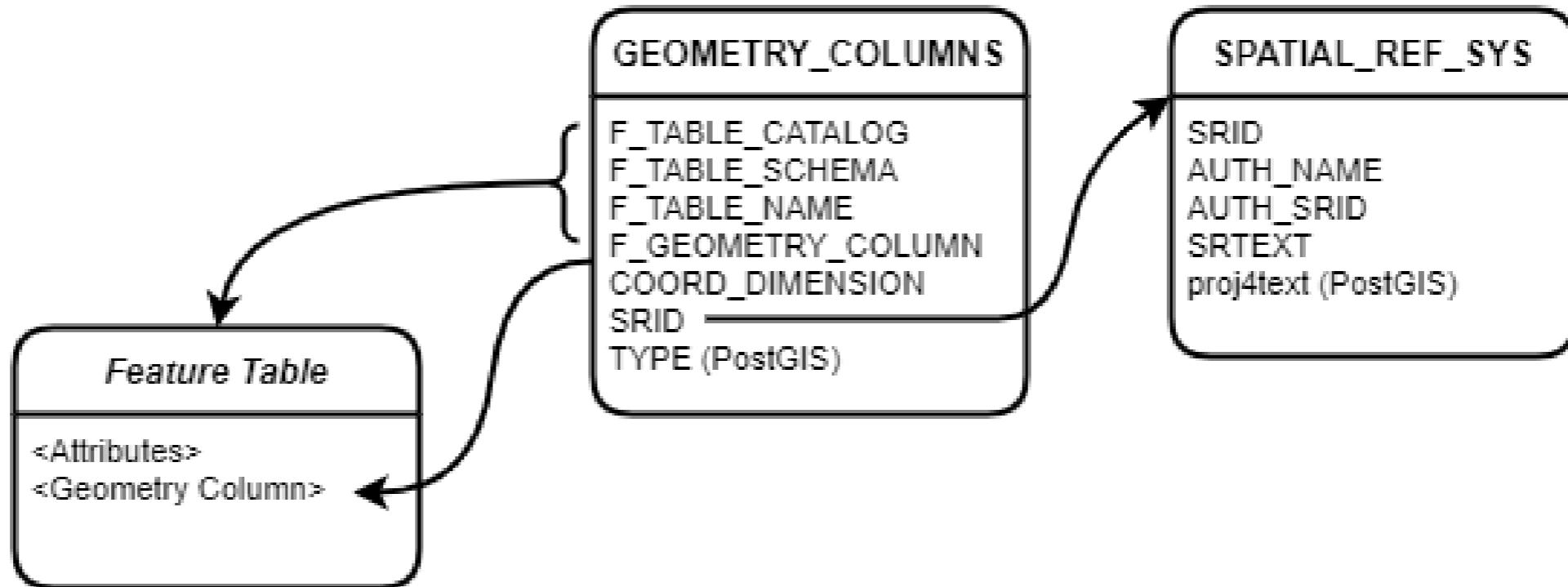
B: LINESTRING (0 0, 1 0, 2 0)

ST_Equals(A, B) == ?

Standards /



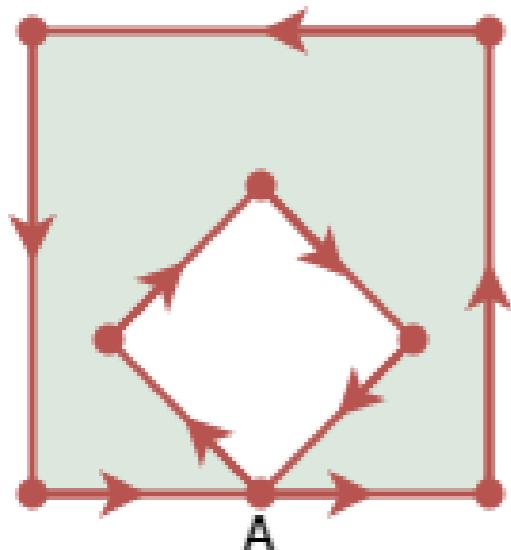
Standards / OGC / Simple Feature / Metatabellen



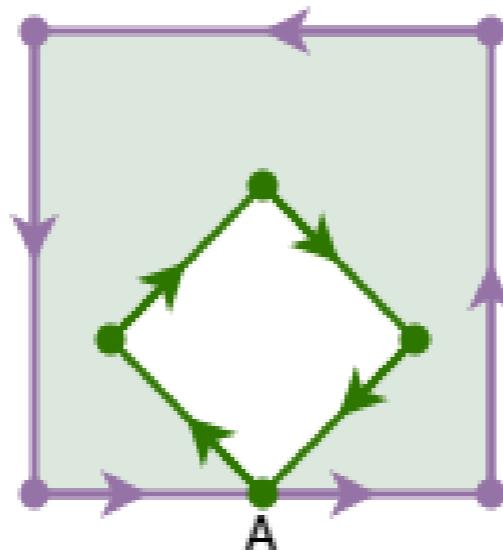
Quelle: nach Abbildung 3 im Abschnitt 6.2.1 in: *OpenG/S Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option*, Version 1.2.1, Open Geospatial Consortium, Inc., 2010.

Realisiert PostGIS GEOMETRY_COLUMNS als View oder als Base Table? ([Quiz](#))

Standards / OGC / Simple Features / Validierung



One ring defines the shape; note that it self-touches at vertex A



Two rings define the shape: inner ring touches exteriorring at A

«Gipfeli Polygon» — ein Ring oder zwei Ringe?

innen ist immer links



Linien dürfen sich selbst schneiden

Standards / weitere Themen

WMS, WMTS, WFS, WCS

Weitere OGC Standards

Esri REST API each iface has two sides (LF)

INTERLIS (und INSPIRE)

XML, JSON, HTTP, etc.

ASCII, Unicode, UTF-8

ISO 8601

[https://ows.terrestris.de/osm/service
?REQUEST=GetMap
&SERVICE=WMS&VERSION=1.3.0
&LAYERS=OSM-WMS&STYLES=
&CRS=EPSG:4326
&BBOX=51.49451,-0.11377,51.53267,-0.06971
&WIDTH=400&HEIGHT=300
&FORMAT=image/png&TRANSPARENT=TRUE](https://ows.terrestris.de/osm/service?REQUEST=GetMap&SERVICE=WMS&VERSION=1.3.0&LAYERS=OSM-WMS&STYLES=&CRS=EPSG:4326&BBOX=51.49451,-0.11377,51.53267,-0.06971&WIDTH=400&HEIGHT=300&FORMAT=image/png&TRANSPARENT=TRUE)

<http://a.tile.openstreetmap.org/15/9798/14664.png>

[http://ows.eox.at/cite/mapserver?service=wcs&version=2.0.1&request
=getcoverage&coverageid=MER_FRS_1PNUPA20090701_124435
_000005122080_00224_38354_6861_RGB](http://ows.eox.at/cite/mapserver?service=wcs&version=2.0.1&request=getcoverage&coverageid=MER_FRS_1PNUPA20090701_124435_000005122080_00224_38354_6861_RGB)

developers.arcgis.com/rest/services-reference
www.interlis.ch
www.json.org
unicode.org und www.unicode.org/charts
en.wikipedia.org/wiki/ISO_8601

charset vs encoding

Isolation und Automation

Isolation: Virtual Environments

<https://www.youtube.com/watch?v=N5vscPTWK0k>

To follow along in Windows:

- start an Anaconda or Miniconda prompt
- to activate: project1_env\Scripts\activate (not source project1_env/bin/activate)
- more Linux → Windows: which → where, ls → dir, cat → type, rm --rf → rmdir /s /q

pip and **virtualenv** sind Python-Klassiker für Paket- und Umgebungs-Management

Im Bereich Data Science eher **conda**: ein Tool für beides (Pakete und Umgebungen)

Isolation: Conda Environments

conda ist Teil der Anaconda-Distribution (und der Miniconda-Distribution)

<https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html>

Default Environment heisst “base” – für jedes Projekt ein eigenes Environment erzeugen!

Wie gross ist so eine Umgebung? (Unix: du -sh ~/miniconda3/envs/*)

Environment löschen: conda env remove -n envname

Tipp: conda env create hat(te?) einen Bug, besser conda create verwenden!

Isolation: Container

Virtual/Conda Environments – lockere Isolation und nur für Python

Virtuelle Maschinen – vollständige Isolation, aber aufwändig und Ressourcen-hungrig

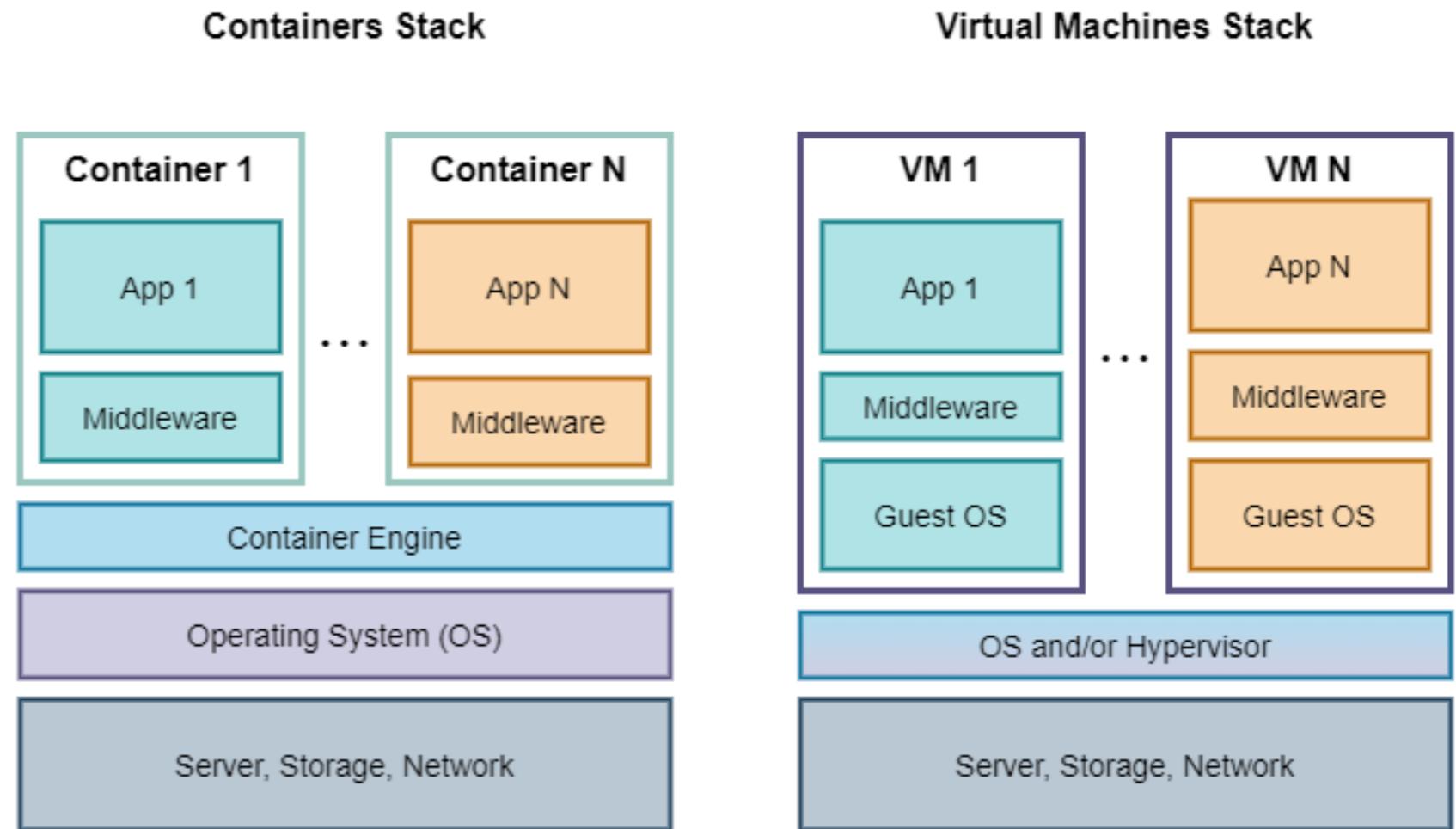
Container – der Trend der Stunde...

- virtualisieren das Betriebssystem: 1 OS-Kern, viele Container
- isolieren das File System: jeder Container hat sein eigenes
- sparen Ressourcen (inkl. Betriebssystem-Lizenzen)
- kontrollierte Zugänge: Mounts (Filesystem) und Ports (Netzwerk)

Docker (www.docker.com) aktuell dominante Container-Technologie

Isolation / Containerisierung

Containers vs. Virtual Machines



Isolation / Containerisierung

Container Image

Image Layers

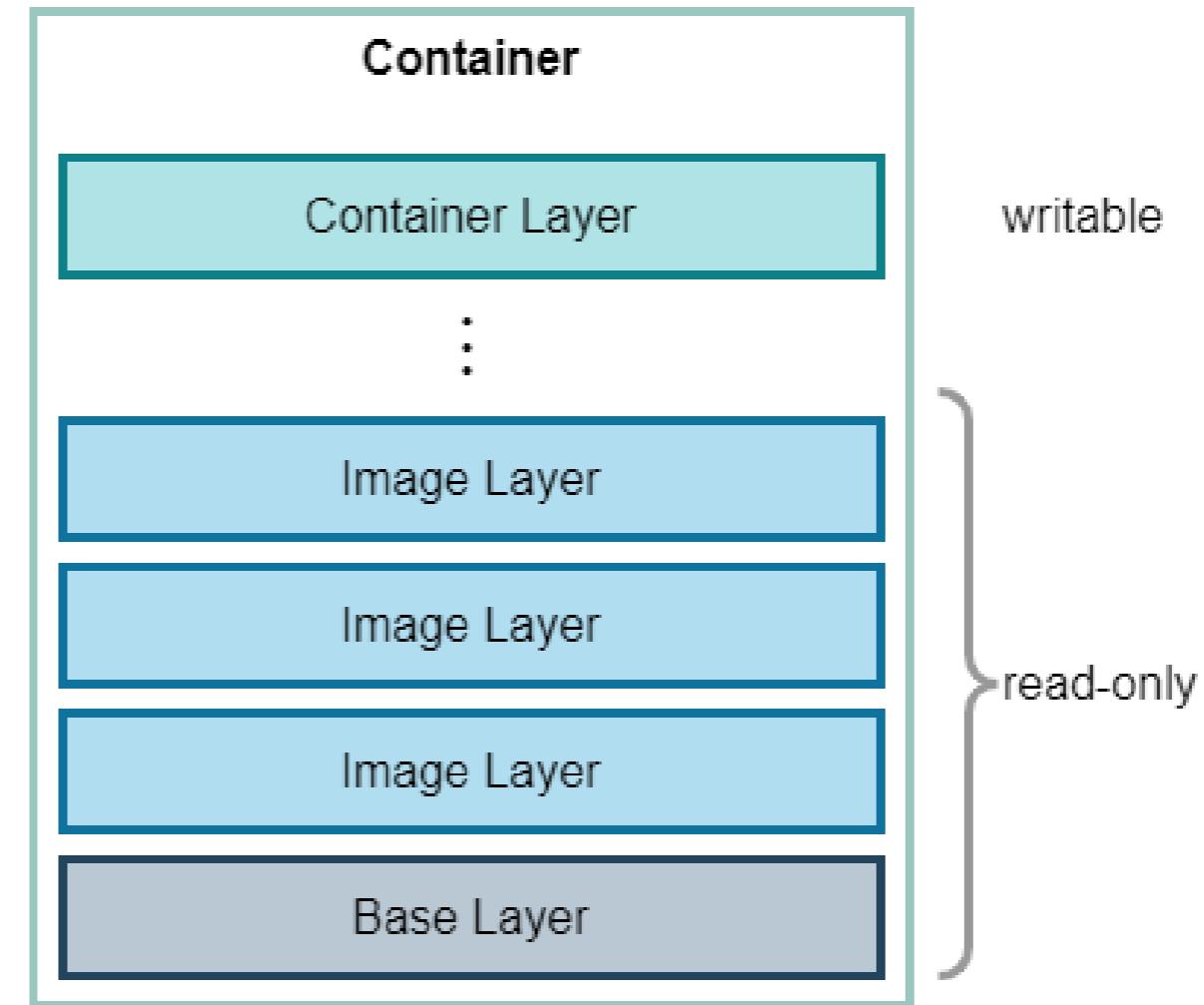
Dockerfile

ADD ...

COPY ...

RUN ...

FROM ...



Isolation / Containerisierung

Container Registry

z.B. hub.docker.com

Unternehmen/Institutionen haben oft ihre eigene

- on premise oder meist in der Cloud

Container Image mit PostgreSQL+PostGIS? ([Übung](#))

- falls mehrere: welches bevorzugen?

Isolation / Containerisierung

Jupyter Docker Stacks

- fertige Container Images für diese beliebte Kombination
- jupyter-docker-stacks.readthedocs.io – go explore! ([Übung](#))

GDS_env

- auf obiger Basis eine Umgebung für die räumliche Analyse
- darribas.org/gds_env – find it on Docker Hub ([Übung](#))
- wer kriegt eines davon lokal zum Laufen? ([Übung](#))

Isolation / Containerisierung

**Erfahrung ist die härteste Lehrerin:
sie gibt dir zuerst die Prüfung,
erst dann den Unterricht.**

Skript Seite 32... gängige Docker-Befehle und Image-Namen

Muster: *registry/user/repository:tag*

Beispiel: *docker.io/postgis/postgis:latest*

Isolation / in SQL

```
CREATE SCHEMA A;  
CREATE TABLE T(...);  
CREATE TABLE A.T(...);  
SELECT * FROM T;  
SET search_path = A, public;  
SELECT * FROM T;
```

Harte Trennung: separate Datenbanken

Weiche Trennung: Konzepte von SQL

Schema (Namensraum)

```
SELECT * FROM information_schema.tables;  
SELECT * FROM geometry_columns;
```

Catalog (Datenbank)

[Demo in pgAdmin](#)

Catalog.Schema.Table

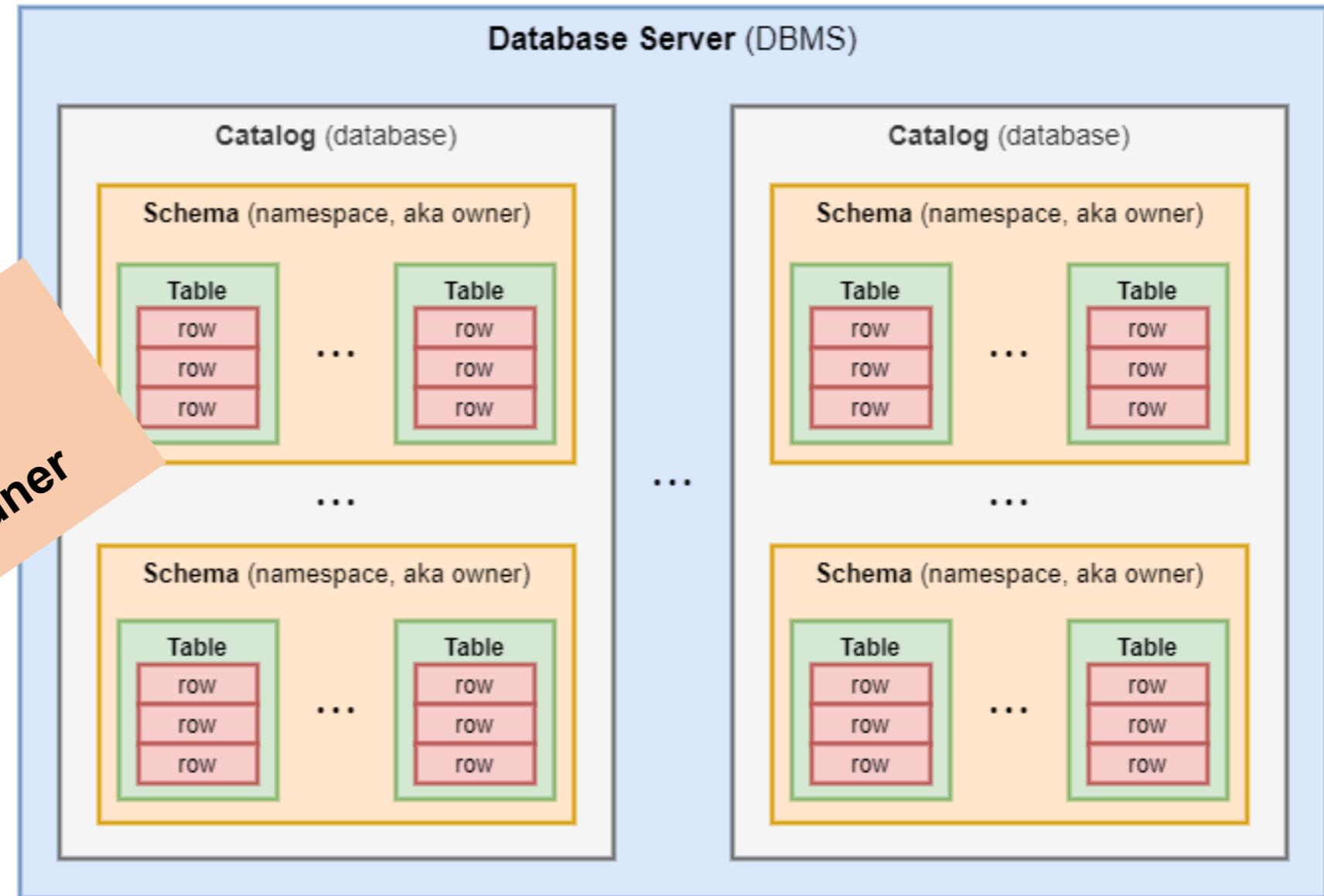
(SQL Server, PostgreSQL)

Schema.Table@Catalog

(Oracle)

Isolation / in SQL

First things first:
jedes Projekt beginnt
mit einem Projektordner



Quelle: nach der Abbildung in <https://stackoverflow.com/questions/7022755>, verändert

Automatisierung



Automatisierung

Warum? – wiederholbar, reproduzierbar – Experimentierfreude...

Notebooks – Dokumentation ist Automatisierung – Automatisierung ist Dokumentation!

Scripting – braucht CLIs (explain: API/CLI)

Make ([Demo](#)) Projects/UseMake

Git (allg. SCM)

Deklarationsdateien (z.B. *Dockerfile*, *compose.yml*, *environment.yml*, *requirements.txt*, ...)

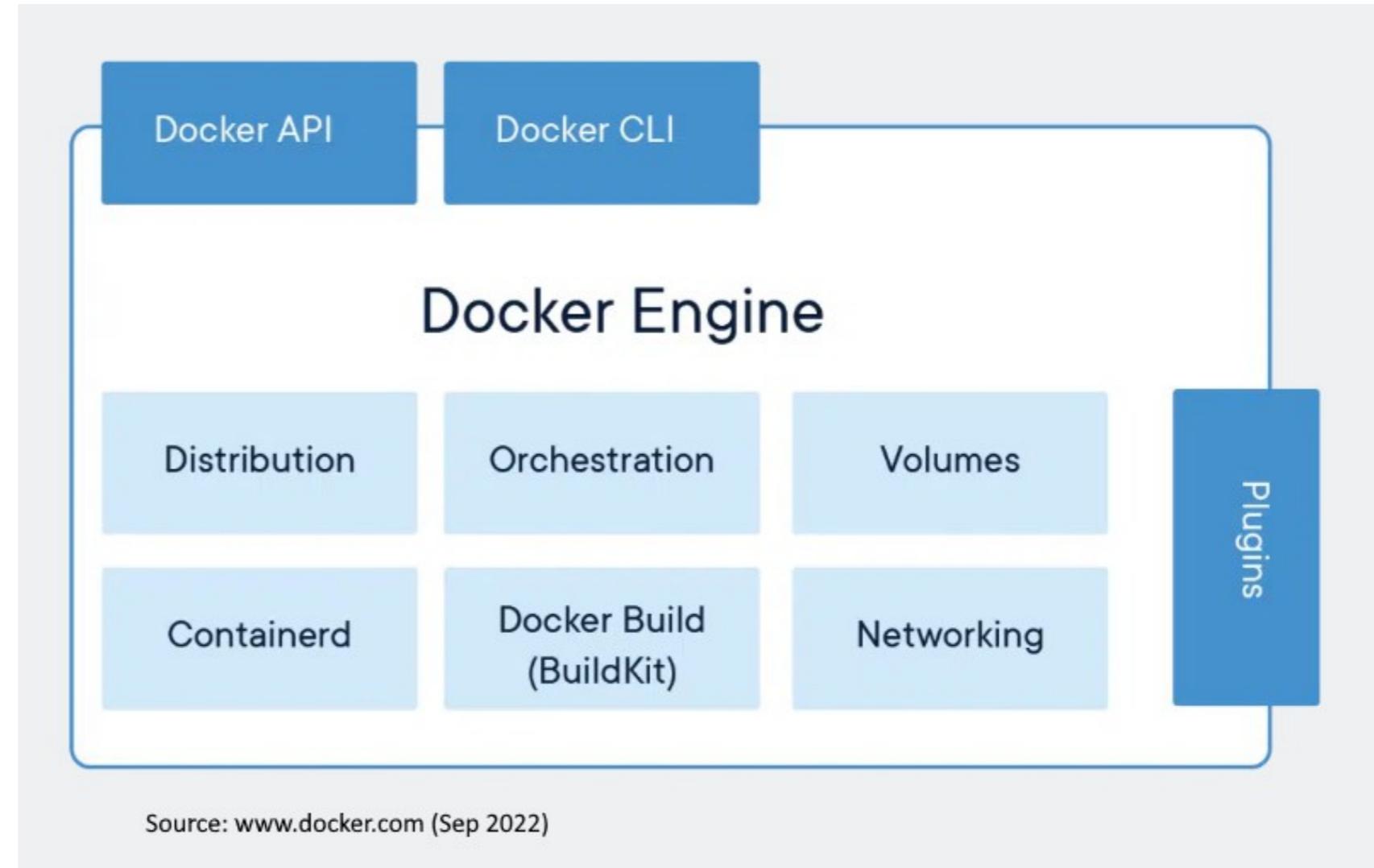
- fast wie Scripting, aber deklarativ: Sollzustand beschreiben (Stichwort DSC, Idempotenz)
- Blick in die Erzeugung von [GDS env](#) ([Demo](#))

Trendkonzept: Infrastructure as Code

API vs CLI am Beispiel Docker Engine

Docker since 2013

Open Container Initiative (OCI)



Infrastruktur in der Produktion

Infrastruktur in der Produktion

explorativ – produktiv

Interview mit E. Mahler

- swisstopo, sizing, tuning
- Spatial Types
- Quaestor
- Gdb Versioning, Gdb Schema Tables
- Bedeutung von Qualität und QA
- Analysen auf Lidar

Ggf. weitere Beispiele

Calculate Line Widths (GP)
Standortsuche Post CH AG

Infrastruktur in der Produktion

Environments

in einer anderen Bedeutung als bisher

Development, Staging, Production

the classic 3 “compute environments”

ArcGIS als Plattform

ArcGIS als Plattform

... bzw. ein Enterprise GIS als Infrastruktur

Anwendung profitiert von GIS-Wissen

Aufbau und Betrieb verlangen allg. IT-Wissen, nicht GIS-spezifisch

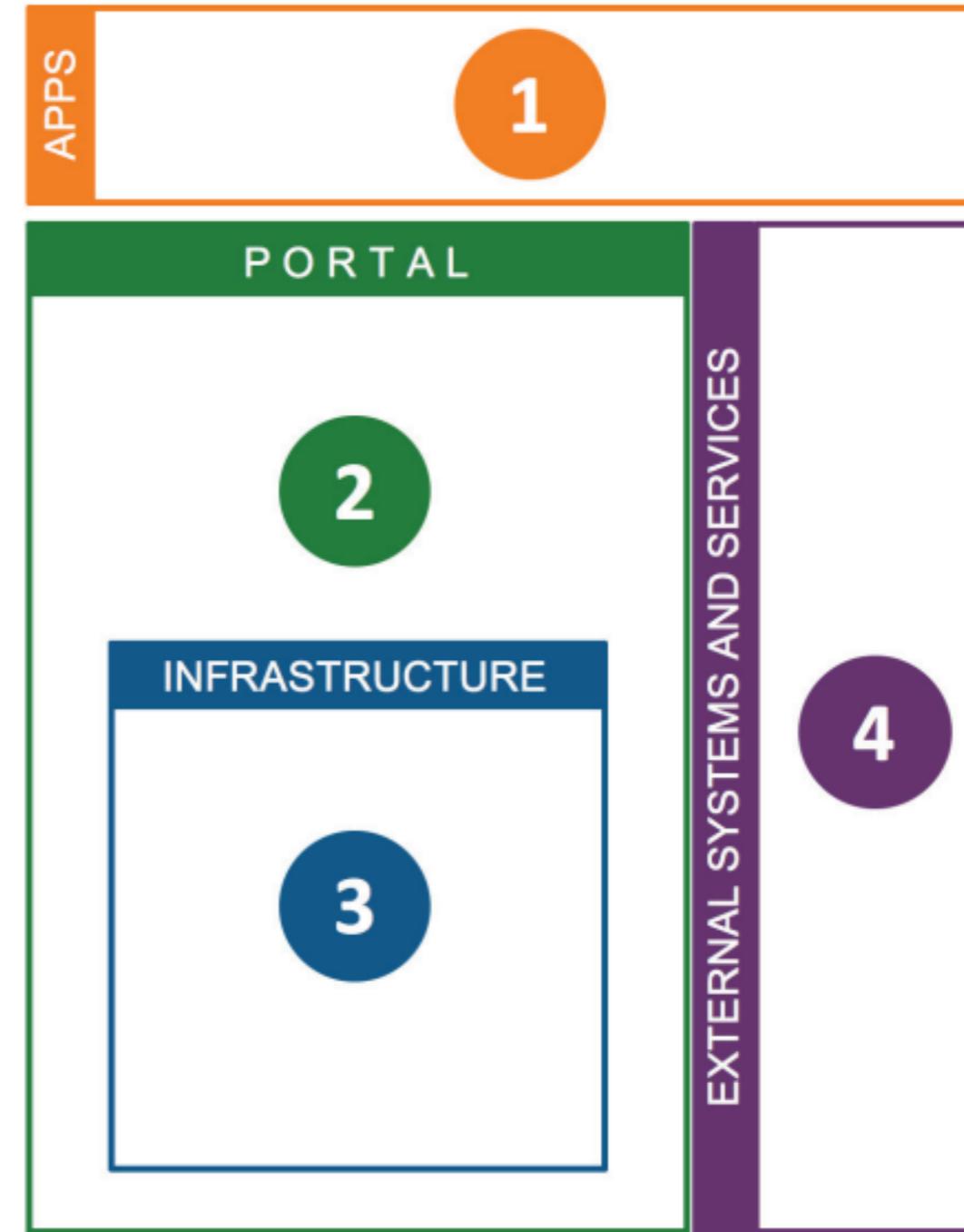
- kann u.U. in der Cloud konsumiert werden (z.B. ArcGIS Online)

Blick auf die Komponenten (Peery 2019) ... Images/EnterpriseGIparts.png



ArcGIS als Plattform

ArcGIS Conceptual Reference Architecture

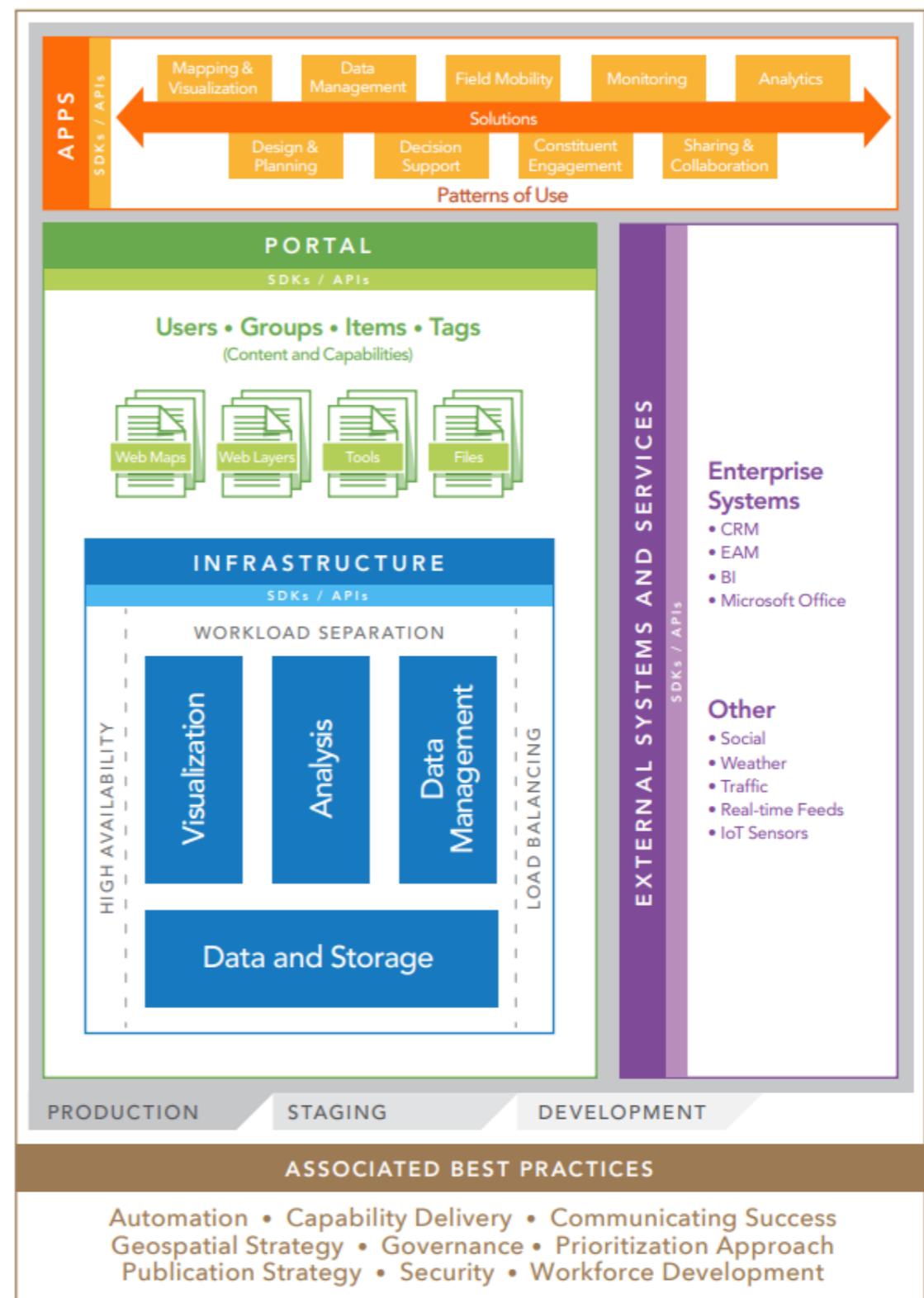


Quelle: Esri 2022, Architecting the ArcGIS System: Best Practices

ArcGIS als Plattform

ArcGIS Conceptual Reference Architecture

- in the cloud
- on premises



Quelle: Esri 2022, Architecting the ArcGIS System: Best Practices

Cloud



Cloud

Definition: ubiquitous, convenient, on-demand, shared, configurable computing resources (NIST)

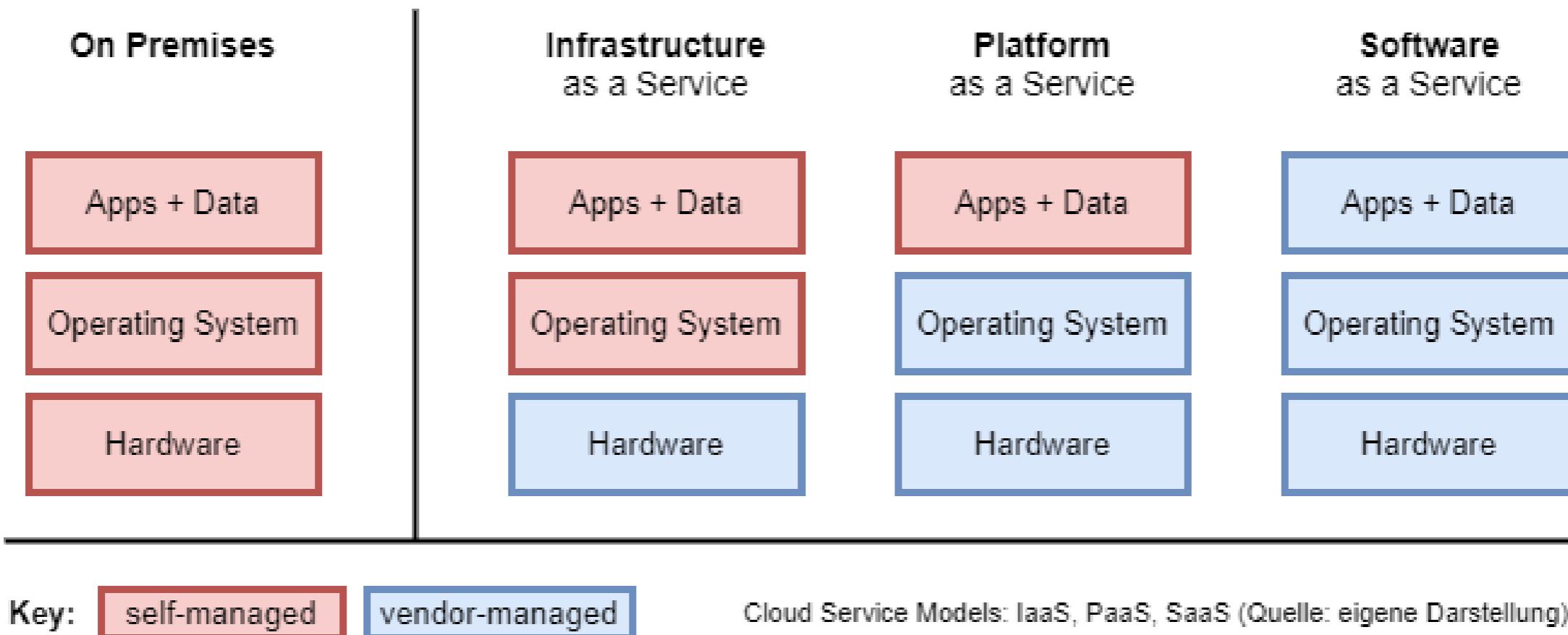
Einfacher: «der Computer eines anderen» (Peery 2019)

Oder: Infrastruktur-Outsourcing

Oder: Umgehung der eigenen IT-Abteilung 😊

Minimalanforderungen: Notebook, Internet-Anschluss, Kreditkarte

Cloud / Service Models



Cloud / Beispiel: ArcGIS Online

Cloud / Beispiel: Binder Project

Binder Project (Jupyter • jupyter.org/binder)

- Input: Git-Repository mit Jupyter Lab/Notebooks
- Output: laufende Instanz auf Cloud-Infrastruktur

Realisiert alle Konzepte der Abschnitte Isolation und Automatisierung:

- ab versioniertem Repo voll automatisch notwendige Infrastruktur bereitstellen
- damit die Analysen in den Notebooks für beliebige Anwender nachvollziehbar

github.com/darribas/bok_chapter_notebooks (Demo, Übung)

Cloud / Beispiel: Azure AKS Quickstart

Optional (falls Zeit und Interesse)

Two Container Application on Azure Kubernetes Service (AKS)

[Projects/Azure_Quickstart](#)

Cloud / CNCF Landscape & Trail Map

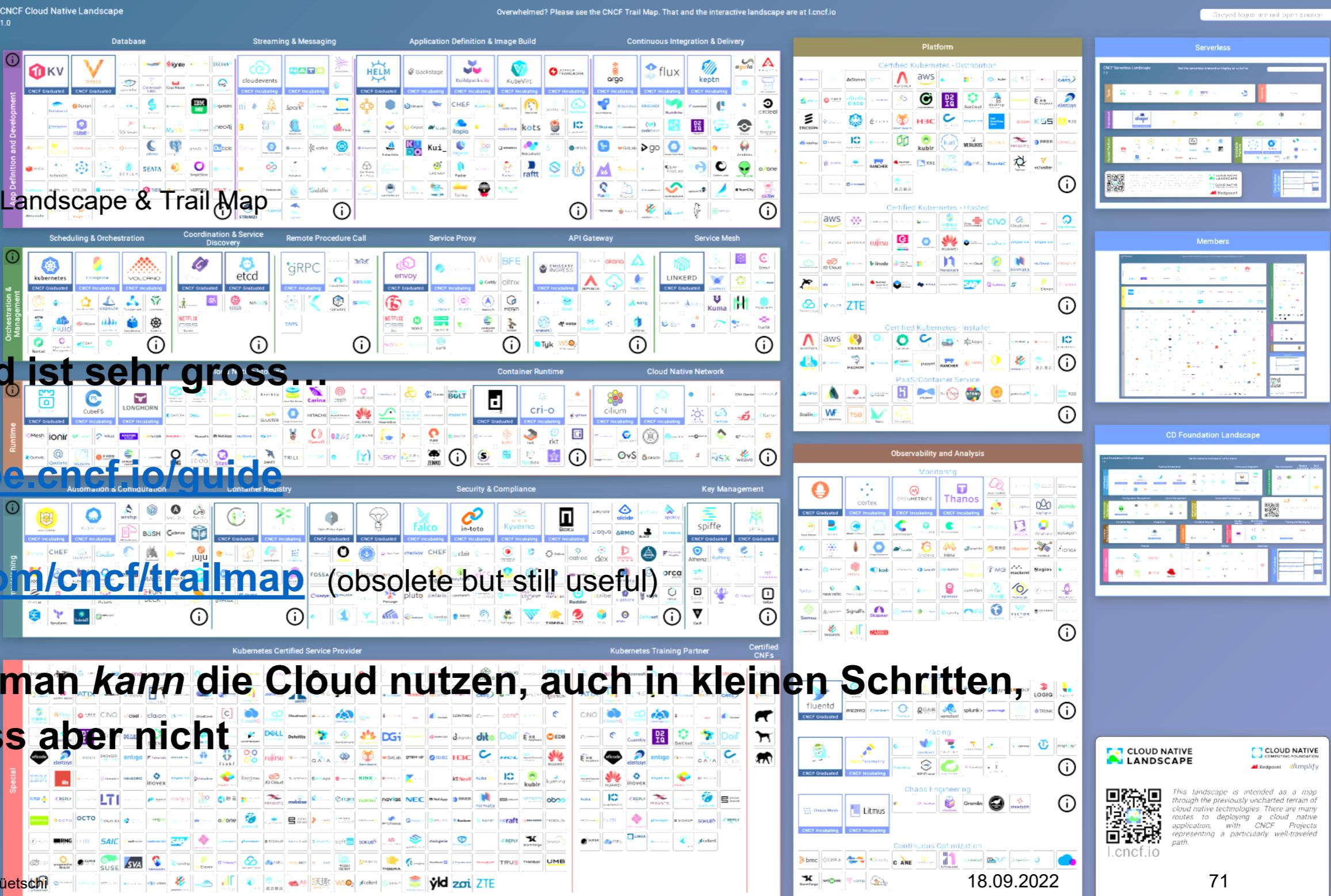
Die Cloud ist sehr gross...

landscape.cncf.io/guide

github.com/cncf/trailmap

(obsolete but still useful)

Wichtig: man kann die Cloud nutzen, auch in kleinen Schritten,
man muss aber nicht



Review



Review

Infrastruktur

- ... die tragenden Schichten unter der Oberfläche
- ... vielfältig, komplex, outsourcebar, unabdingbar

From “iron age” to “cloud age”

Review

Prinzipien (Morris 2015)

- ... systems can be easily reproduced (confidence, not fear)
- ... systems are disposable (robust software on unreliable hardware)
- ... systems are consistent (principle of least surprise)
- ... processes are repeatable (resist one-off changes – script)
- ... design is always changing (make changes frequently in small steps)

Review

«Cloud Native» waren wir nicht ganz

- zu viel lief noch lokal – und das ist auch ok

Blick ins Glossar

Aus der Literatur (siehe Skript)

- für Python/Jupyter sehr spannend: pythongis.org, geographicdata.science
- für GIS Computing Platforms: gistbok.ucgis.org
- für die Cloud: Tutorials der jeweiligen Provider
- für Plattformen: Webseiten der Hersteller (z.B. www.esri.com)

Besten Dank und viel Erfolg!
—Urs-Jakob Rüetschi

