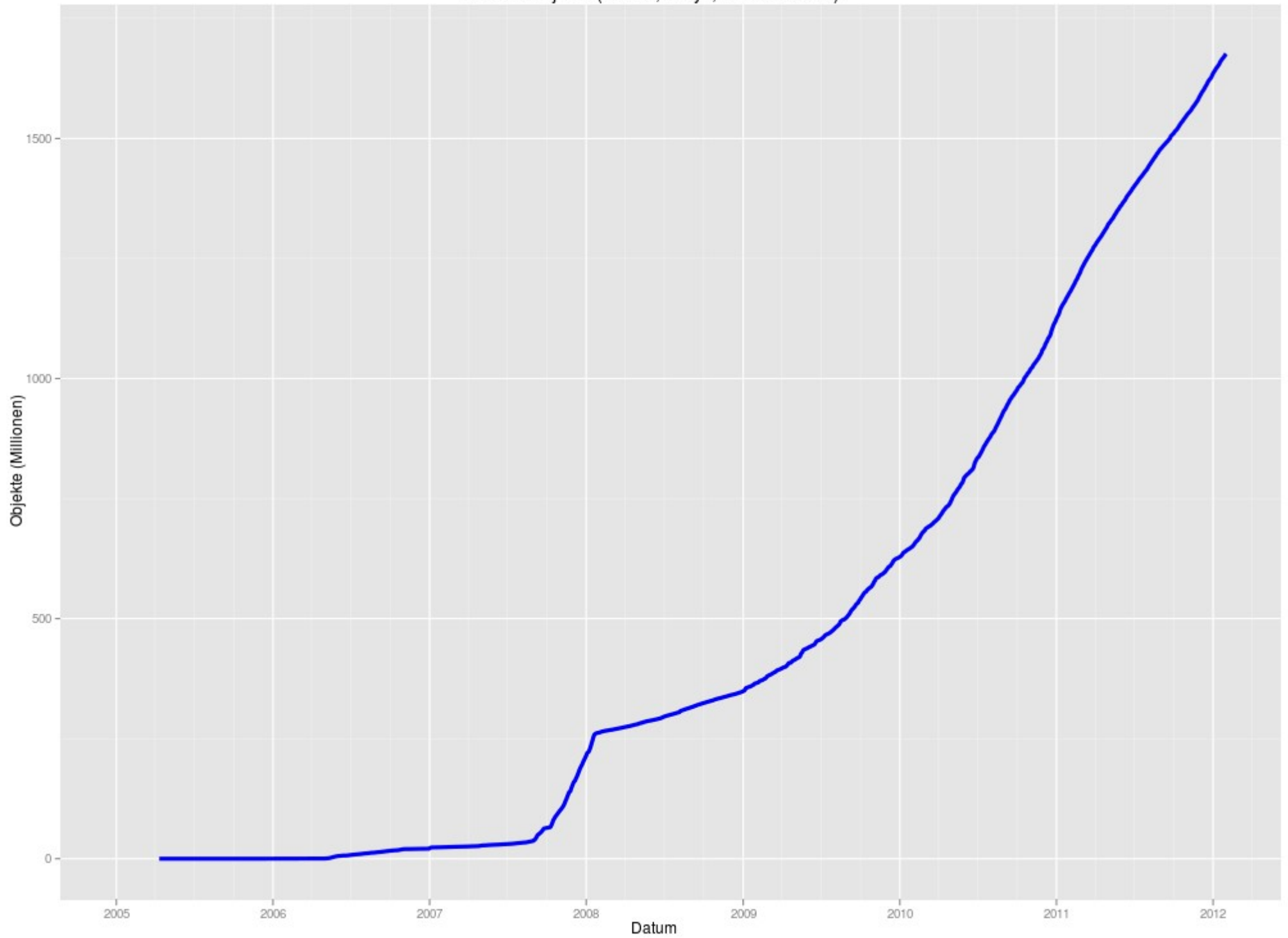




Das Osmium-Framework

Jochen Topf

Anzahl Objekte (Nodes, Ways, und Relations)







INTERNATIONAL BESTSELLER

I CAN MAKE YOU

THIN

The Revolutionary
System Used
by More Than
3 Million
People

Includes
Guided
Hypnosis
CD



PAUL McKENNA







Osmium

Osmium is ein
schnelles und flexibles
C++- und Javascript-Toolkit
für das Arbeiten mit OSM-Daten.

Grundlegende Objekte

Node

Way

Relation

Area

Position

Bounds

Tag

RelationMember

...

Benutzung dieser Objekte

```
node.id()
```

```
node.timestamp_as_string()
```

```
node.position().lon()
```

```
way.nodes().first().ref()
```

```
relation.members()
```

```
node.tags().add("highway", "service")
```

```
way.tags()[1].key()
```

Geometrie-Objekte

Point

LineString

Polygon

MultiPolygon

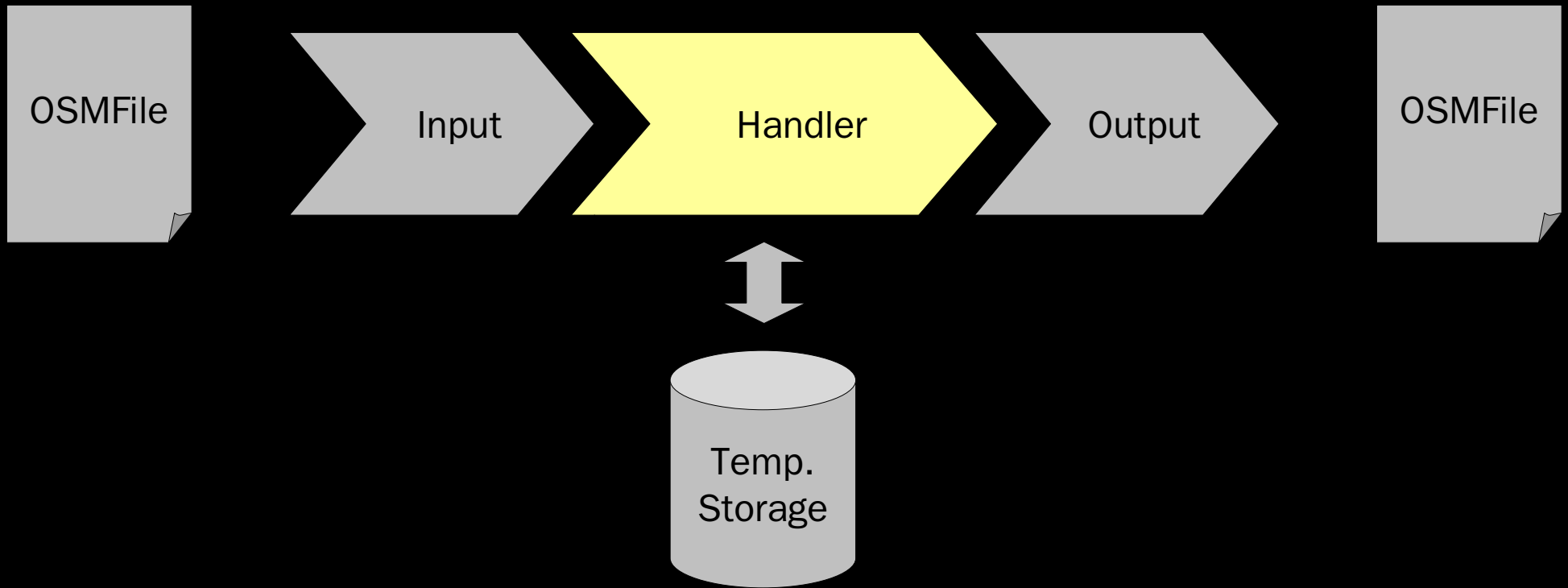
„Anbindung“ von OGR, GEOS, Shapelib

Geometrie-Objekte

```
Point p(node);  
cout << p.as_WKT() << "\n";  
cout << p.as_HexWKB() << "\n";  
p.create_geos_geometry();  
p.create_ogr_geometry();  
p.create_shp_object();
```

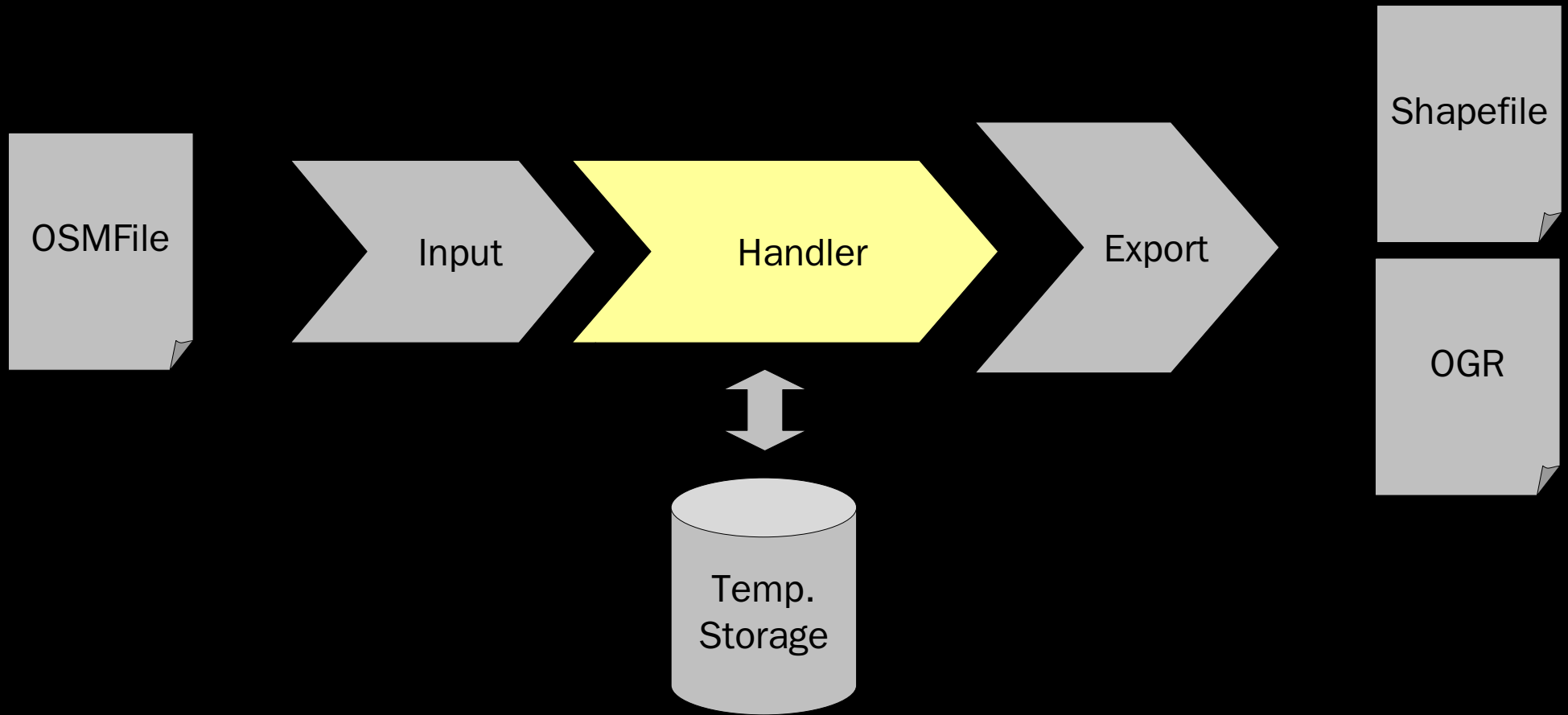
```
LineString line(way, true);  
MultiPolygon mp(area);
```

Architektur I



Für Konvertierer und Filter

Architektur II



Zum Export von OSM-Daten

OSMFile

Lesen von: Datei, stdin oder URL.
Schreiben nach: Datei oder stdout.

XML oder PBF.

Komprimiert oder Unkomprimiert.

Mit oder ohne History.

OSM-Daten (.osm) oder Changes (.osc).

Handler-Callbacks

init(meta)

node(node)

way(way)

relation(relation)

final()

Beispiel-Handler

```
class MyHandler : public Osmium::Handler::Base
{
public:

    void node(const shared_ptr<Node>& node)
    {
        cout << "node id=" << node->id() << "\n";
        cout << "    lon=" << node->lon() << "\n";
        cout << "    lat=" << node->lat() << "\n";
    }
}
```

Beispiel: Hauptprogramm

```
#include <osmium.hpp>

int main(int argc, char* argv[]) {
    Osmium::init();

    Osmium::OSMFile infile(argv[1]);
    MyHandler handler;
    infile.read(handler);
}
```

Mitgelieferte Handler

Debug

CoordinatesForWays

Multipolygon

RangeFromHistory

ObjectStore / ApplyHandler

Storage

ById

ObjectStore

CoordinatesForWays-Handler

- liest Positionen aus den Nodes
- schreibt Positionen in die Ways

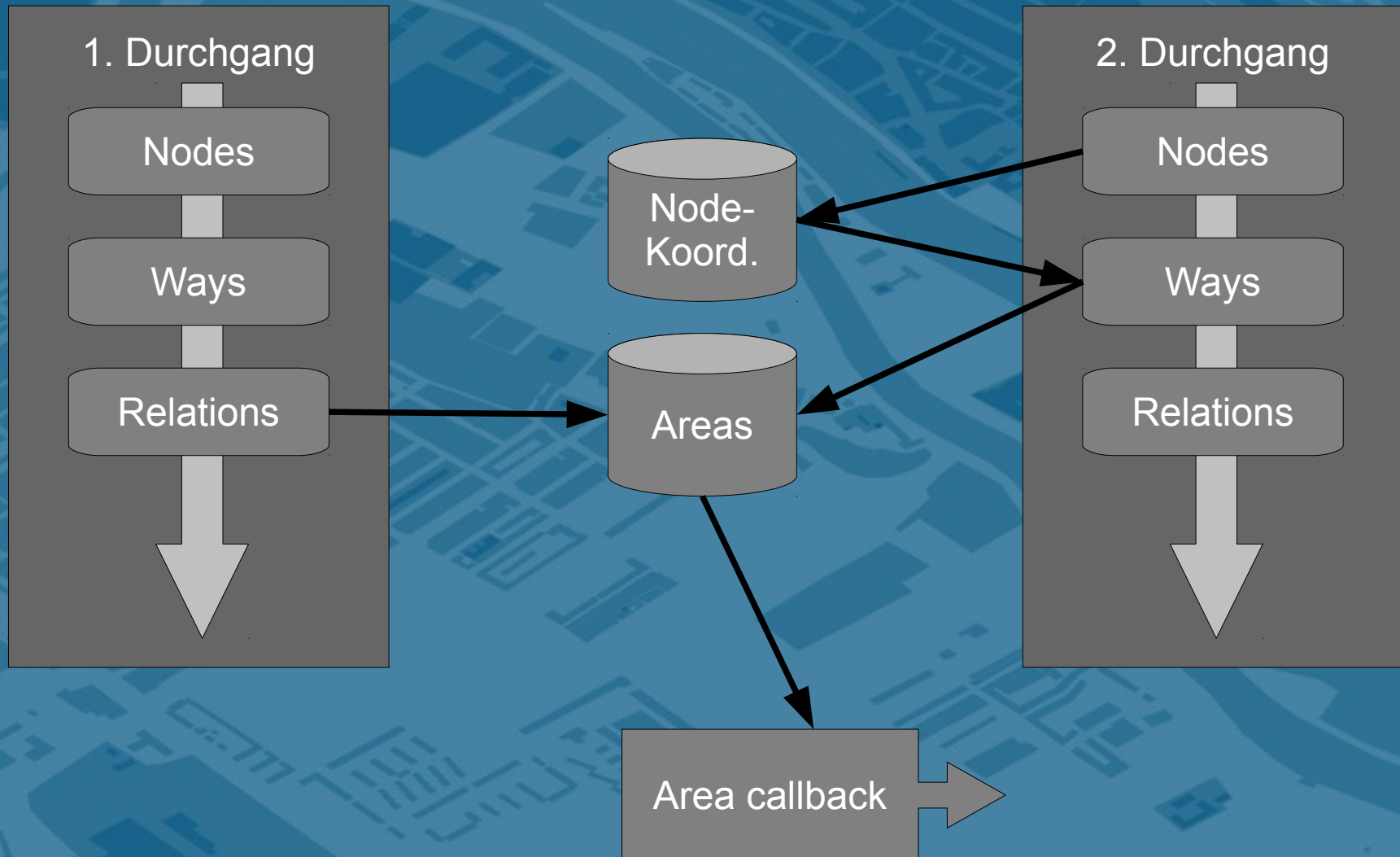
Zwischenspeicher:
Mmap oder SparseTable

Multipolygone

aus Ways und Relations
(type=multipolygon/boundary)

Verschachtelte Multipolygone.
Berührende innere Ringe.
Ignoriert falsche Roles (inner/outer).
Korrigiert manche kaputten Geometrien.

Multipolygone zusammensetzen



Kompilieren

Header-only Library.

Relativ sauberer, dokumentierter Code.

Kompiliert ohne Warnings

Mit GCC (g++) und LLVM (clang).

Ein paar (nicht genug) Unit-Tests.

Optionen:

`-D OSMIUM_WITH_JAVASCRIPT/GEOS/OGR/SHPLIB`

Betriebssysteme

Entwicklung unter Linux

Kompiliert unter Mac OSX

Windows ?

Beispiel-Programme

osmium_debug
osmium_convert
osmium_toshape
osmium_toogr
osmium_store_and_debug
nodedensity

...



taginfo.openstreetmap.org

Diverse Statistiken zu
60 Mio. verschiedenen Tags
auf 1,5 Mrd. Objekten.

Läuft in circa 2 Stunden durch.

Braucht circa 6 GB RAM.

But wait...

...there is more

C++ ist nicht jedermanns Sache.

Google V8 Javascript-Engine

Relative einfach in C++-Code einzubinden

Kompiliert Javascript

Sehr schnell

Open Source

Osmium Javascript-Handler

Callbacks in den Javascript-Code des Users

Zugriff auf grundlegende Objekte

Zugriff auf Export-Funktionen

osmjs

Enthält Javascript- und Multipolygon-Handler

```
osmjs -j objects.js planet.osm.pbf
```

```
osmjs -m -l array -j objects_2pass.js planet.osm.pbf
```

Unterstützung für History-Daten

Lesen/Schreiben

Changes anwenden

Zeitliche Extrakte

Vielen Dank an Peter Körner

The End

wiki.osm.org/wiki/Osmium

osmiumapi.openstreetmap.de

github.com/joto/osmium

Jochen Topf

jochen@topf.org