The Internet Communication Engine

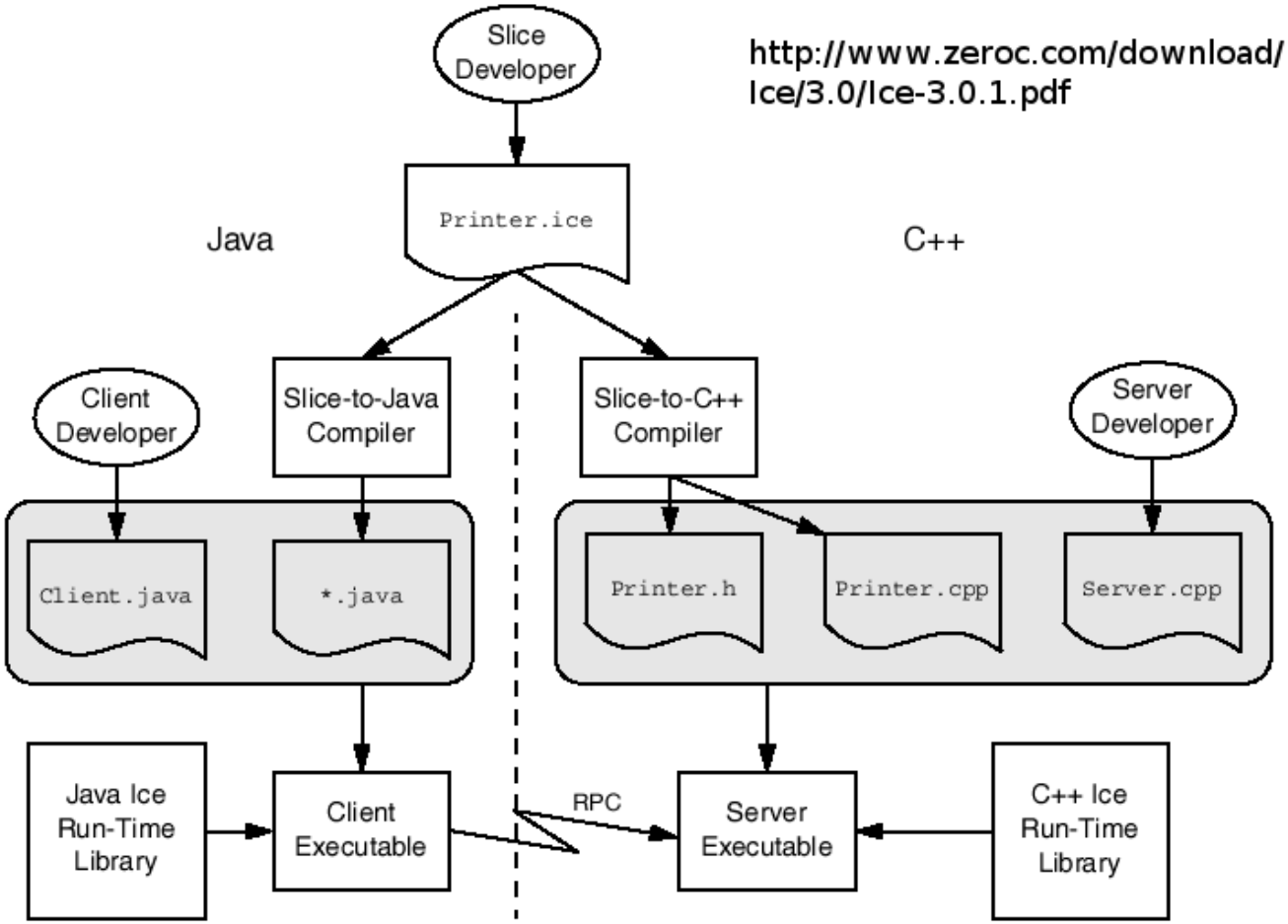
Alexander Bernauer
alex@ulm.ccc.de

8. Mai 2006

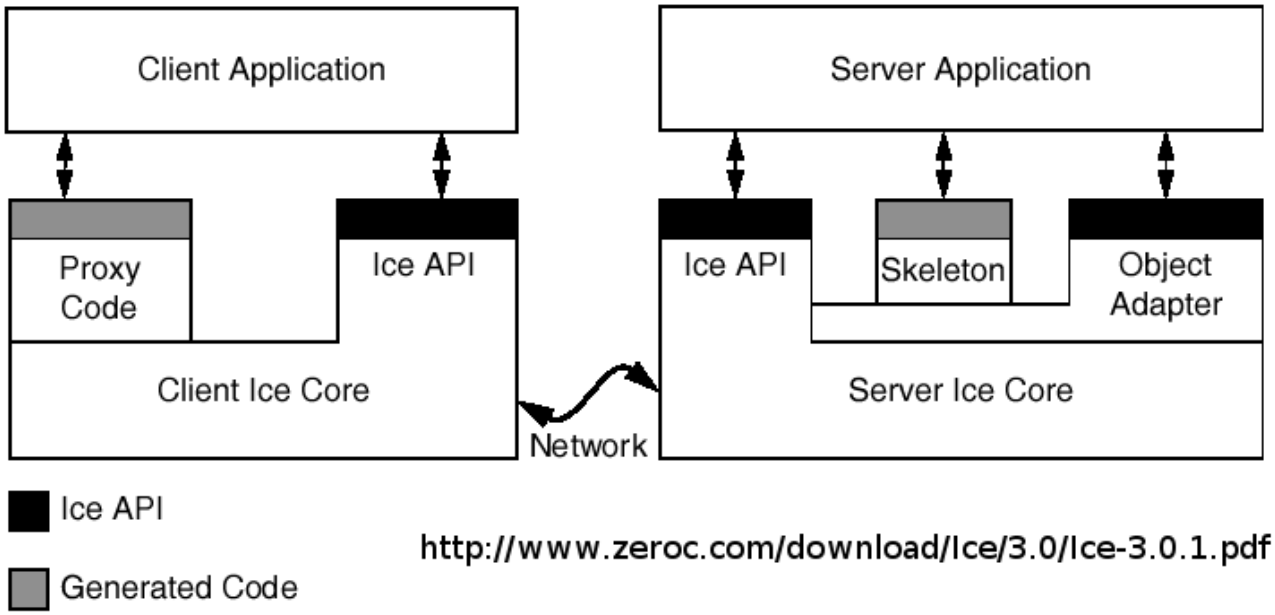
Motivation

- RPC
- objektorientiert
- transparent
- plattformunabhängig
- sprachunabhängig

Funktionsprinzip



Architektur



http://www.zeroc.com/download/Ice/3.0/Ice-3.0.1.pdf

Aufrufsemantiken

- At-most-Once
- synchron oder asynchron
- [Batched]-Oneway
- [Batched]-Datagram

Alternativen

- CORBA
- RMI
- WebService
- ...

Slice

- Specification Language for Ice
- plattform- und sprachunabhängige Sprache zur Beschreibung von Schnittstellen

module

```
1 module ProjektName {
2     // Deklerationen
3 };
```

- verschachteln
- mehrmals öffnen

Basistypen

Typ	Bereich	Größe
`bool`	`false`, `true`	$\geq$ 1 bit
`byte`	-128 bis 127 bzw. 0 bis 255	$\geq$ 8 bit
`short`	-2^{15} bis $2^{15}-1$	$\geq$ 16 bit
`int`	-2^{31} bis $2^{31}-1$	$\geq$ 32 bit
`long`	-2^{63} bis $2^{63}-1$	$\geq$ 64 bit
`float`	IEEE single-precision	$\geq$ 32bit
`double`	IEEE double-precision	$\geq$ 32bit
`string`	alle Unicode Zeichen	variabel

struct

```
1 struct TimeOfDay {
2     short hour;
3     short minute;
4     short second;
5 };
```

- Auflistung von Definitionen, keine Typdeklarationen
- keine Vererbung
- dürfen nicht leer sein

sequence

```
1  sequence < int > IntSeq;
```

dictionary

```
1 dictionary < int, string > Map;
```

enum

```
1 enum Fruit { Apple, Pear, Orange };
```

- Keine Festlegung der Konstanten möglich

interface

```
1 interface Clock {
2     TimeOfDay getTime();
3     void setTime(TimeOfDay time);
4 };
```

- mehrere Rückgabewerte über out-Parameter
- kein Überladen möglich

interface - Teil 2

```
1 interface Clock {
2     nonmutating TimeOfDay getTime();
3     idempotent void setTime(TimeOfDay time);
4 };
```

interface - Teil 3

```
1 interface B { /* ... */ };
2 interface C1 extends B { /* ... */ };
3 interface C2 extends B { /* ... */ };
4 interface D extends I1, I2 { /* ... */ };
```

- immer Diamand
- Bezeichner dürfen nicht mehrdeutig sein
- implizite Vaterklasse `Ice::Object`

Proxies

```
1 interface Clock {
2     nonmutating TimeOfDay getTime();
3     idempotent void setTime(TimeOfDay time);
4 };
5
6 interface Bootstrap {
7     Clock* initClock();
8 };
```

- pass-by-reference Semantik

exception

```
1 exception IllegalValueException { };
2
3 interface Clock {
4     nonmutating TimeOfDay getTime();
5     idempotent void setTime(TimeOfDay time) throws IllegalValueException;
6 };
```

- Vererbung möglich

class

- Hybrid aus `struct` und `interface`
- Einfachvererbung (`extends`)
- Implementierung von Interfaces (`implements`)
- Selbstbezug möglich

Beispiele

Streams

- Serialisierung und Deserialisierung von ICE Typen

Facets

- Alternative Interfaces für ein ICE Objekt
- Hilfreich für Versionierung

SSL

- Zertifkat und Konfiguration
- Adapter-Endpoint mit SSL

Dienste

- Freeze
- Glacier2
- IceBox
- IceStorm
- IcePatch2

Bashing

- ICE ist wirklich brauchbar
- Aber manche Dinge sind echt nervig

Slice

- Präprozessor
  - `#ifdef`-Guards
  - zyklische Abhängigkeiten
- keine Vorwärtsdeklaration für Typen

Slice - Teil 2

- Case-insensitiv
- Belegung von Namensräumen
- Verbot von Wiederverwendung

Slice - Teil 3

- Metainformationen
- Vererbung von Metainformationen

C++-Mapping

- obligatorisches Reference-Counting
  - Objekte dürfen nicht auf den Stack allokiert werden
  - automatisches Löschen von Servant-Objekten bei Shutdown des Communicators
- extra Gargabe-Collector für Zyklen

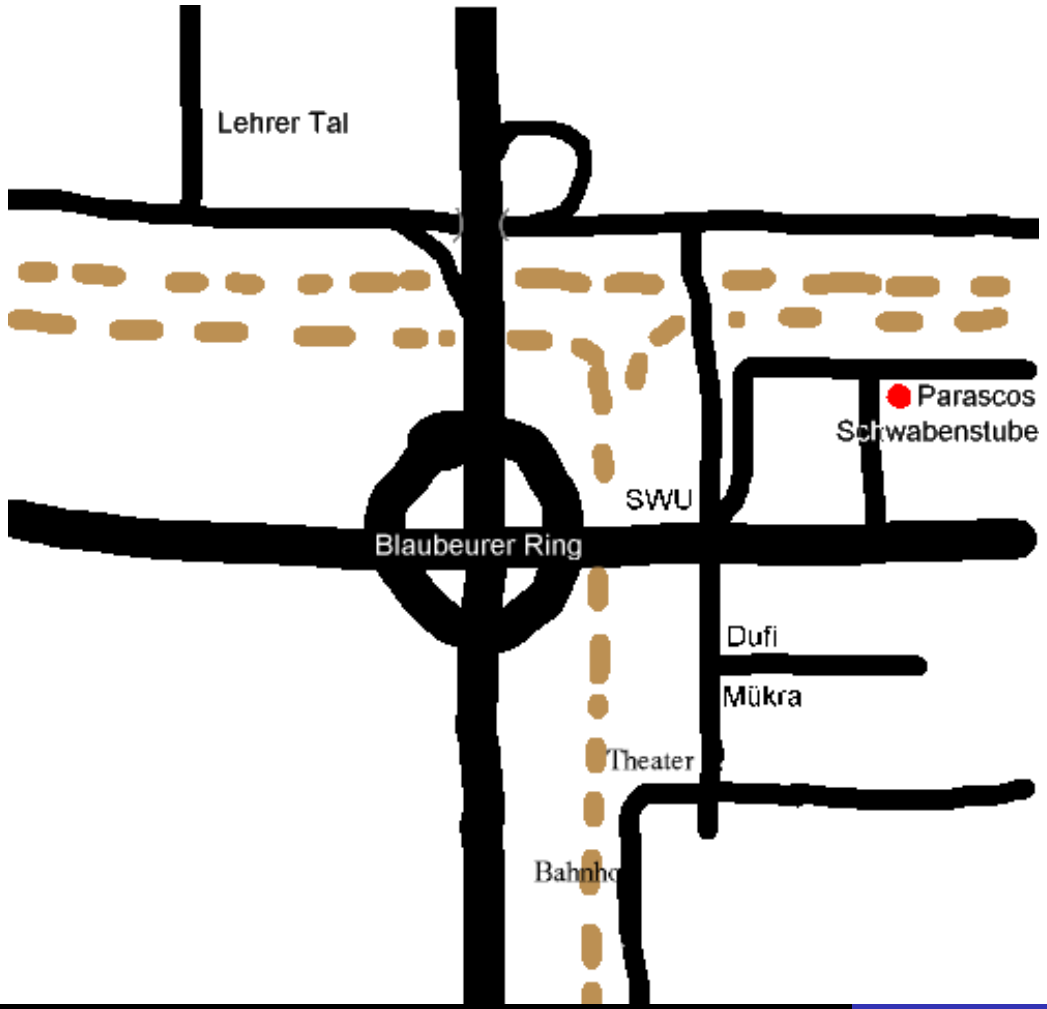
Links

- ZeroC http://www.zeroc.com
- OMG http://www.omg.org

verwendete Software

- Debian GNU/Linux
- Vim
- PDF-LaTeX
- LaTeX-Beamer
- X-PDF
- ion3
- TeX-ify
- GNU make

Parasco


Lehrer Tal
Parascos
Schwabenstube
SWU
Blaubeurer Ring
Dufi
Mükra
Theater
Bahnho