COMARCH

Towards simplicity

Arkadiusz Niemiec

Telco BSS R&D Department

Cracow Erlang Factory Lite, 2012

Agenda

- Short introduction to Comarch
- Service control module
- Credit control application
- Previous solution
- Proof of concept
- Migration to Erlang OTP
- The benefits
- Summary

Comarch Group facts and figures

- **Public company founded in 1993**
- **Organic growth over the past 18 years**: a portfolio of in-house products
- **Nearly 3500 employees worldwide** (including almost 1000 in the Telecom Business Unit)
- **Over 3000 successfully completed projects**
- **Global presence:** Western & Eastern Europe, Middle East, USA, Latin America
- **Diversified product and service portfolio:**

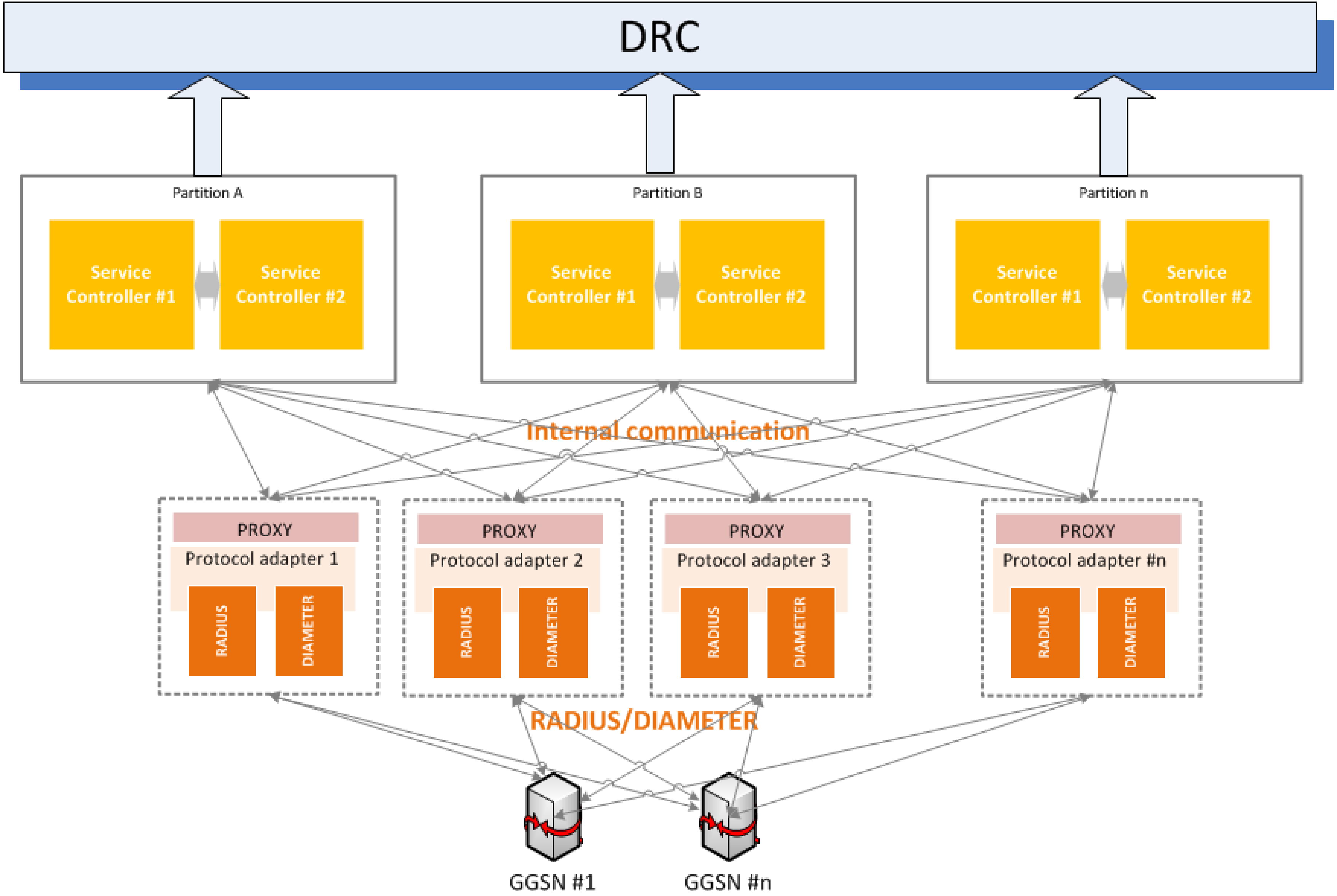
Telecommunications Finance Trade&Services Government ERP

Service Control Module

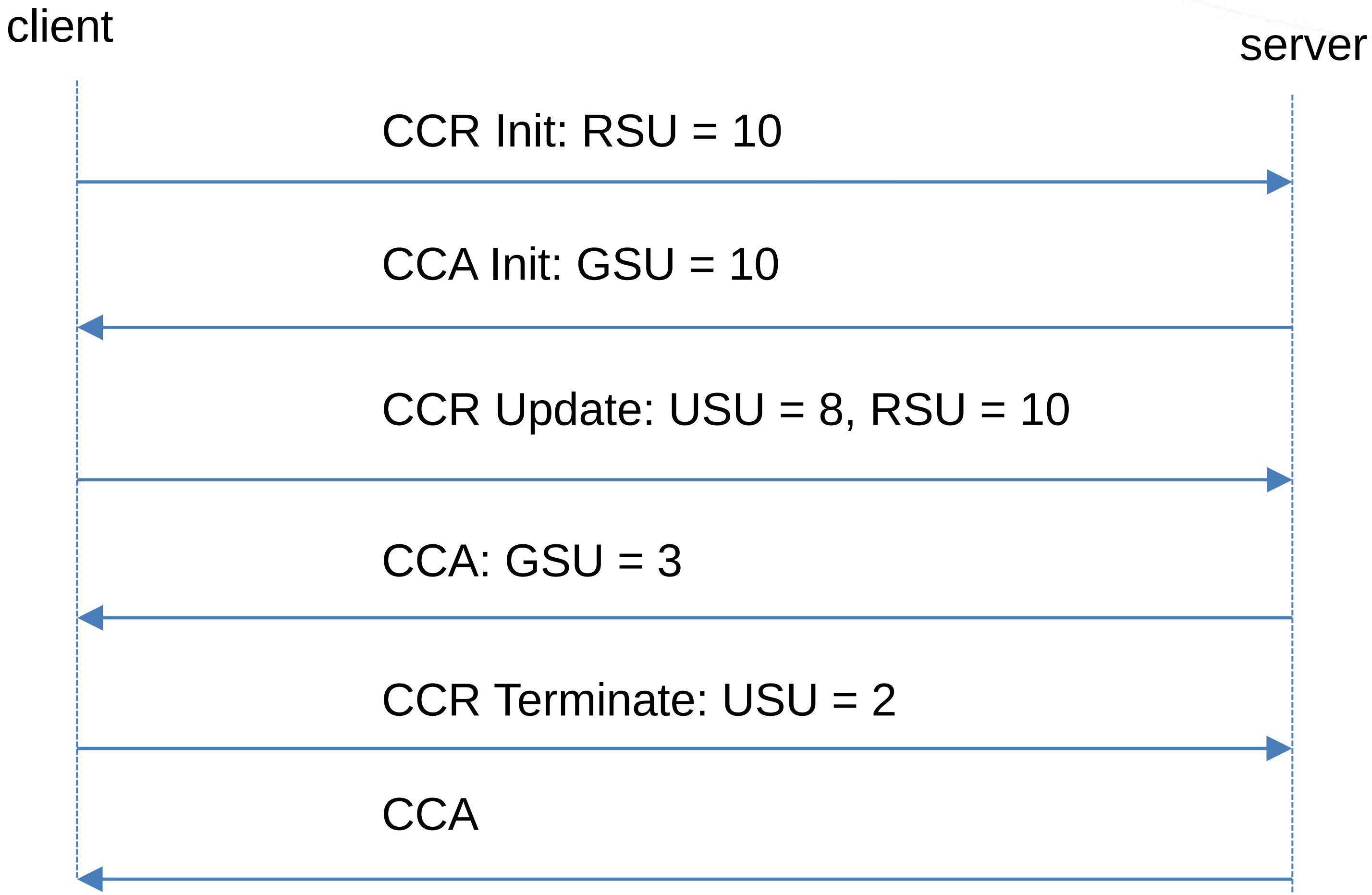
- **Basic roles of SCM:**
  - **AAA:** **A**uthentication, **A**uthorization and **A**ccounting
  - **Service control** (managing sessions existing on the hardware i.e. GGSN)
  - **PCRF:** **P**olicy and **C**harging **R**ules **F**unction

- **Selected functionalities**
  - volume thresholds
  - location limits
  - fraud detection

Trigger → Event → Action


DRC
Partition A
Service Controller #1
Service Controller #2
Partition B
Service Controller #1
Service Controller #2
Partition n
Service Controller #1
Service Controller #2
Internal communication
PROXY
Protocol adapter 1
RADIUS
DIAMETER
PROXY
Protocol adapter 2
RADIUS
DIAMETER
PROXY
Protocol adapter 3
RADIUS
DIAMETER
PROXY
Protocol adapter #n
RADIUS
DIAMETER
RADIUS/DIAMETER
GGSN #1
GGSN #n

Credit control in a nutshell



client

server

CCR Init: RSU = 10

CCA Init: GSU = 10

CCR Update: USU = 8, RSU = 10

CCA: GSU = 3

CCR Terminate: USU = 2

CCA

Creating real-time components

- Nodes discovery
- Replication
- Takeover
- On-line upgrades
- Auto healing

What was the problem then?

We created very sophisticated framework in C++ to address high availability needs

Technical and organizational chalenges

- Sweat and blood to achieve stability of the system
- Very expensive using of 3rd party components
- Defensive programming

- Development was always under stress, huge responsibility
- The lack of understanding C++ from fresh graduated programmers

The problem was always solved in different domain

Proof of concept

- The diameter stack
  - Diameter credit control application
  - Integration with OCS
  - Fully configurable

- Total cost: ~45md
- 4564 LOC, without unit tests

We did a lot of a good useless job (**R14B03)**

COMARCH

Migration chalenges

- Availability
- Horizontal scalability
- Session replication
- Consistency
- Zero-downtime during upgrades
- Low latency

The benefits

- Robustness of Erlang VM
- Smaller code
- No defensive programing
- Erlang console
- Location transparency
- Mnesia
- Easy to maintain

Summary

- Cost savings
- Improved creativity
- Shorten time-to-market
- Simplified internal architecture
- Problem oriented language

Thank you

Arkadiusz.Niemiec@comarch.com