

A Web Service Gateway for SMS-based Services

*Giuseppe Attardi, Daniele Picciaia,
Antonio Zoglio*
Dipartimento di Informatica
Università di Pisa

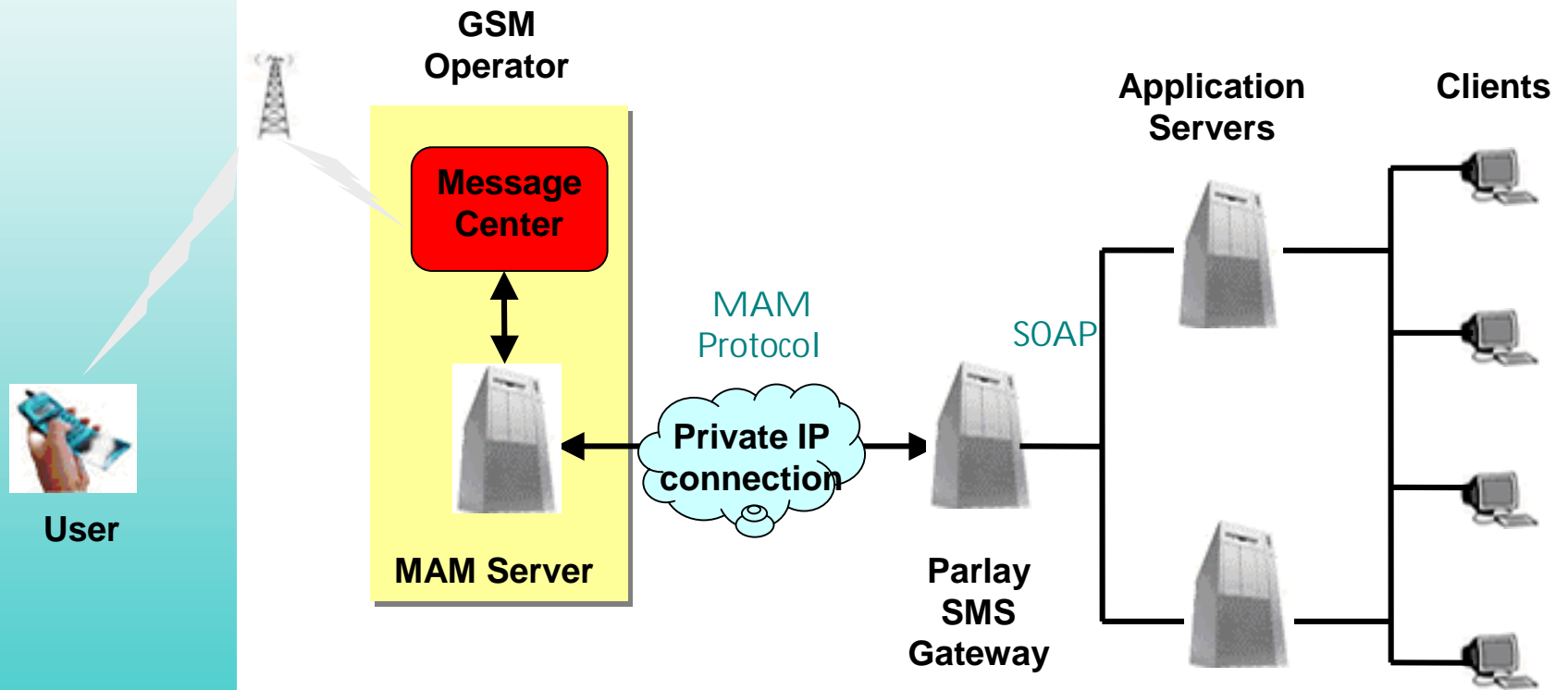
Motivation

- **bridge between telephony applications and Web applications**
- **expose telephony services as XML Web Services**

Protocols

- **Parlay X Web Services**
- **SMS Forum**
 - **SMPP (Short Message Peer to Peer)**
 - **MMAP (Mobile Message Access Protocol)**
- **Vodafone service:**
 - **MAM (Messaggistica Aziendale Mobile)**

Network setting



Software architecture

- **XML Web Service**
- **WSDL**
- **SOAP**
- **Pattern subscribe/notify**

Two-way Web Services

- **Send Service:**

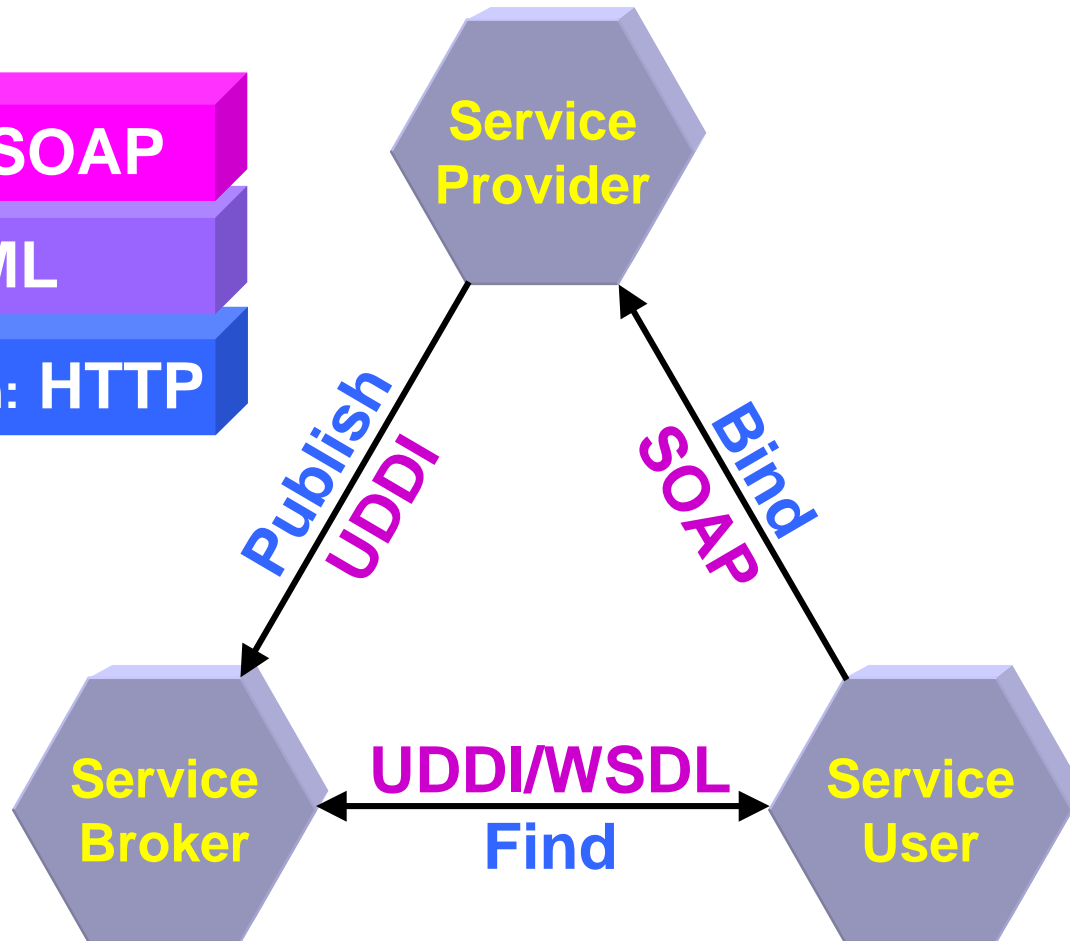
- RequestIdentifier sendSms(destAddressSet, senderName, charging, message)

- **Notify Service:**

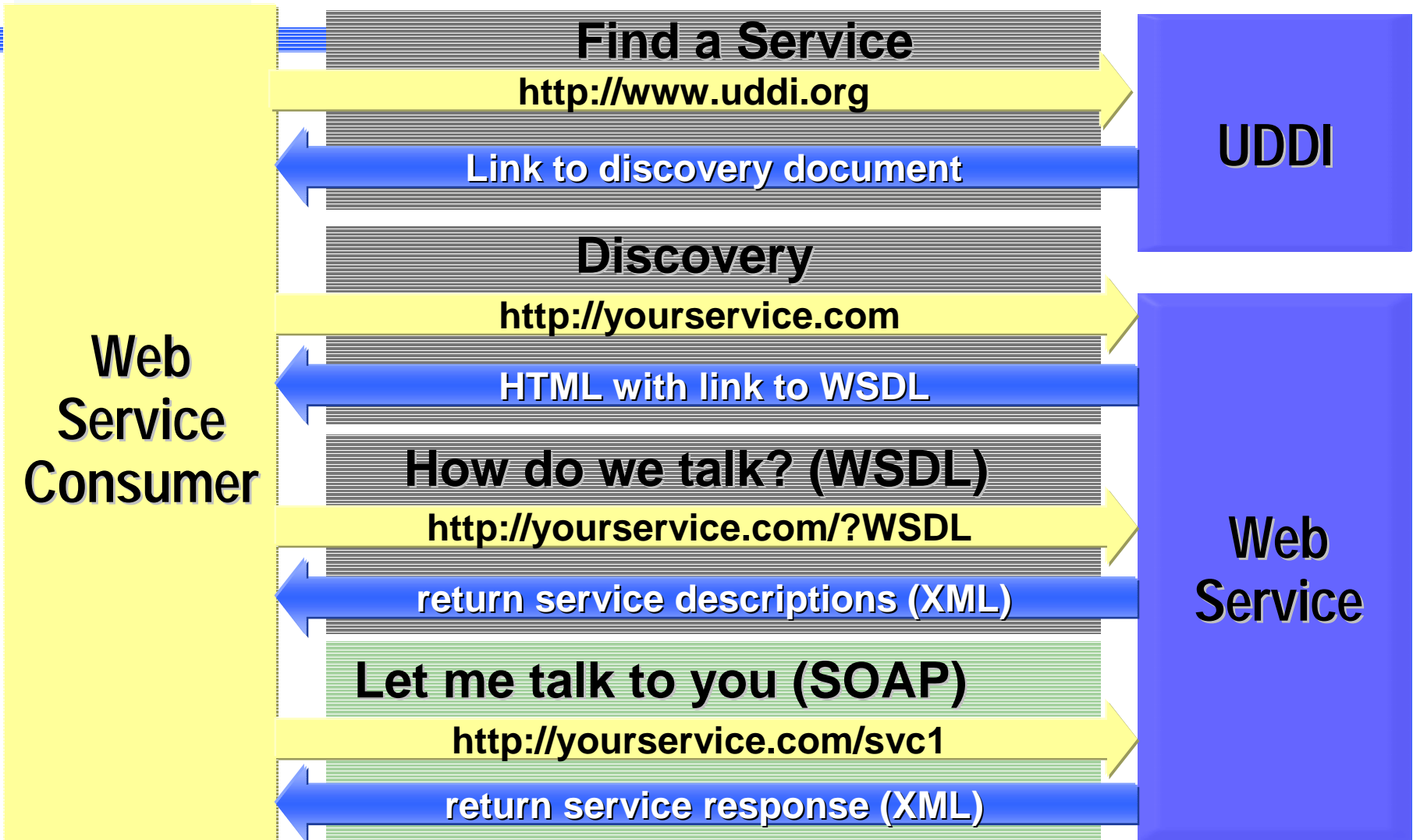
- void notifySmsReception(RegistrationIdentifier, smsServiceActivationNumber, SenderAddress, message)

Web Service Architecture

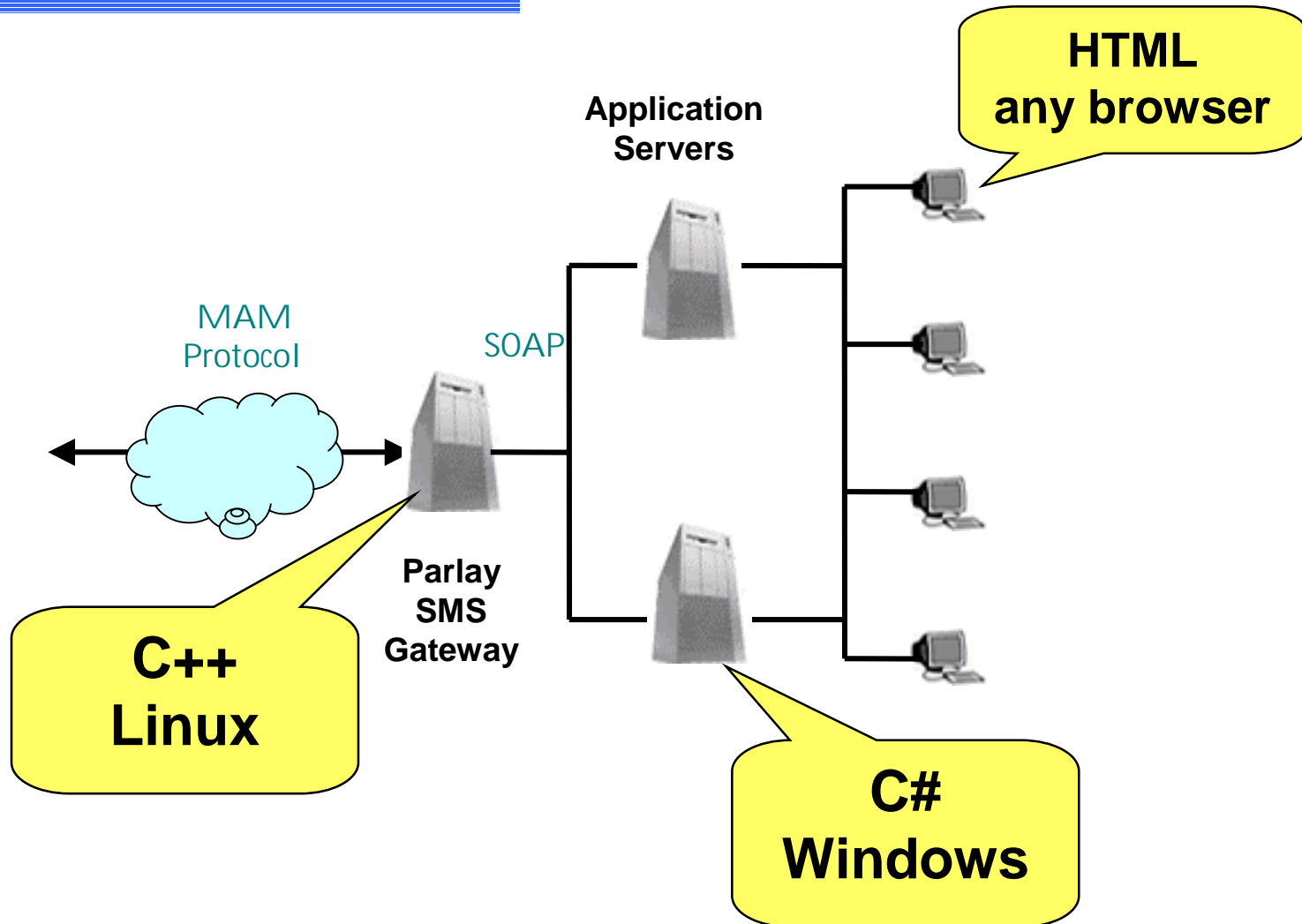
Interactions: **SOAP**
Data: **XML**
Communication: **HTTP**



Web Services Protocols



Interoperability



gSOAP

- **C++ SOAP compiler**
- **From WSDL to .h headers and stub/proxy code**
- **Utilities:**
 - **wsdl2h (generates .h from WSDL)**
 - **soapcpp2 (generates stub/skeleton from .h)**
- **High performance:**
 - **1500 call/sec (gSOAP, C++)**
 - **370 req/sec (Apache Axis, Java)**

WSDL

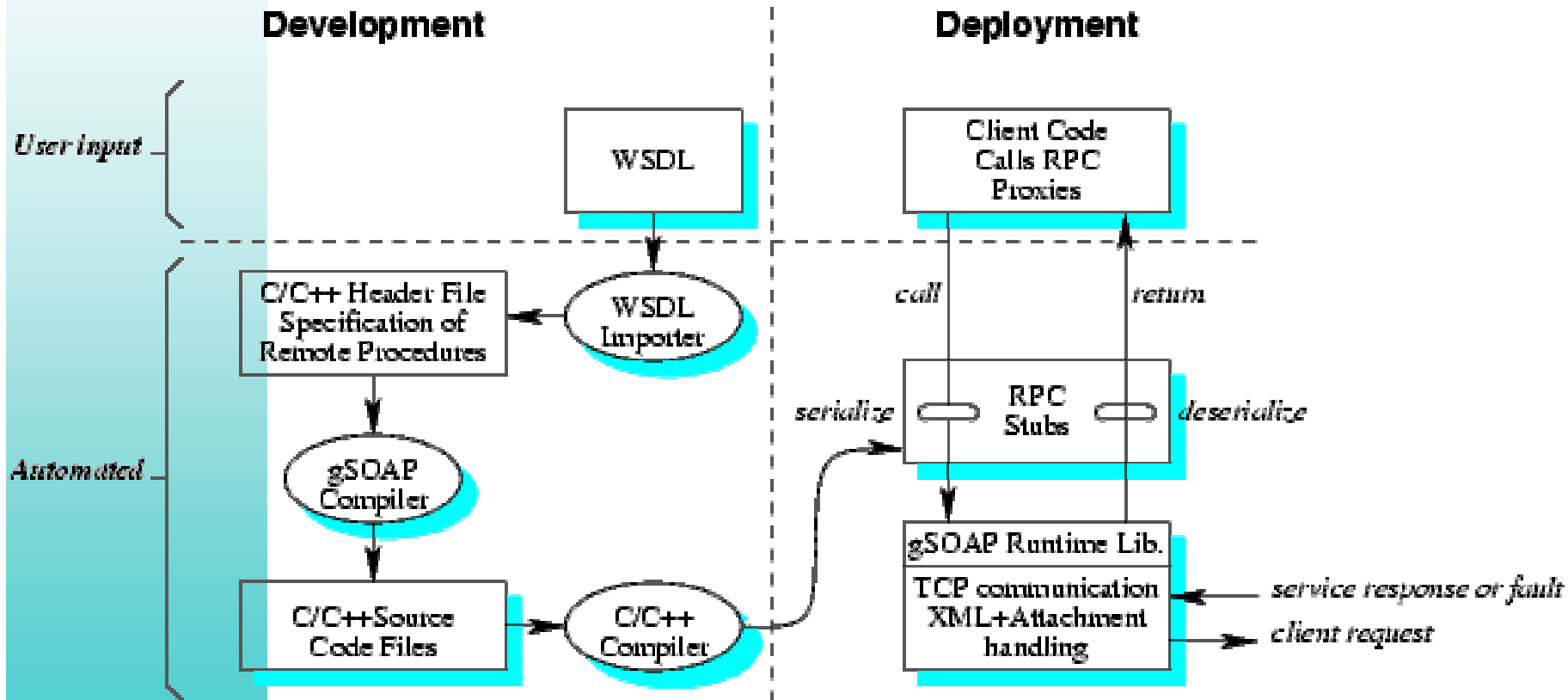
```
<message name="sendSmsRequest">
  <part name="destAddressSet"
    element="tns:destAddressSet"/>
  <part name="senderName"
    element="tns:senderName"/>
  <part name="charging"
    element="tns:charging"/>
  <part name="message"
    element="tns:message"/>
</message>

<message name="sendSmsResponse">
  <part name="result"
    element="tns:SmsIdentifier"/>
</message>
```

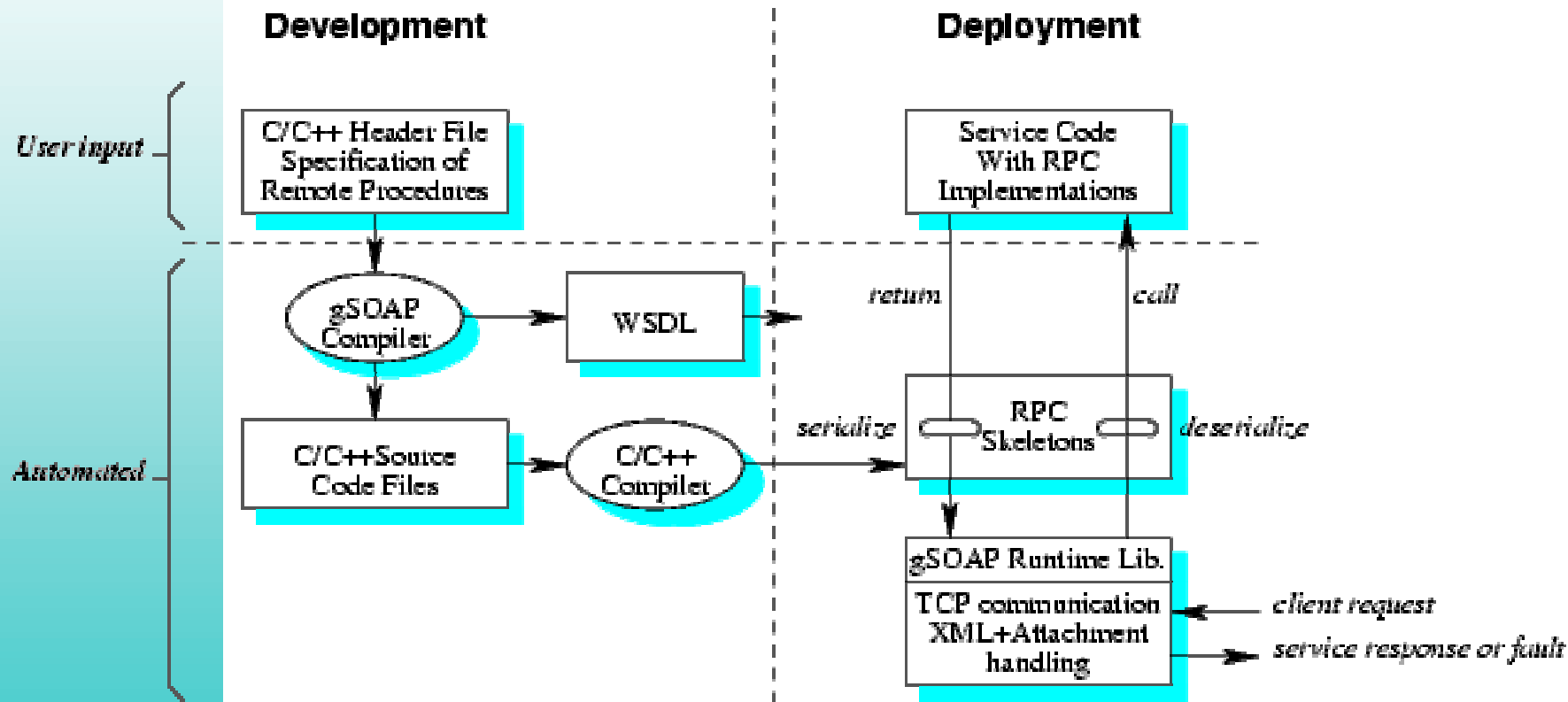
wSDL2h output

```
int sendSms(ArrayOfEndUserIdentifier*  
            destAddressSet,  
            string senderName,  
            string charging,  
            string message,  
            string& SmsIdentifier);
```

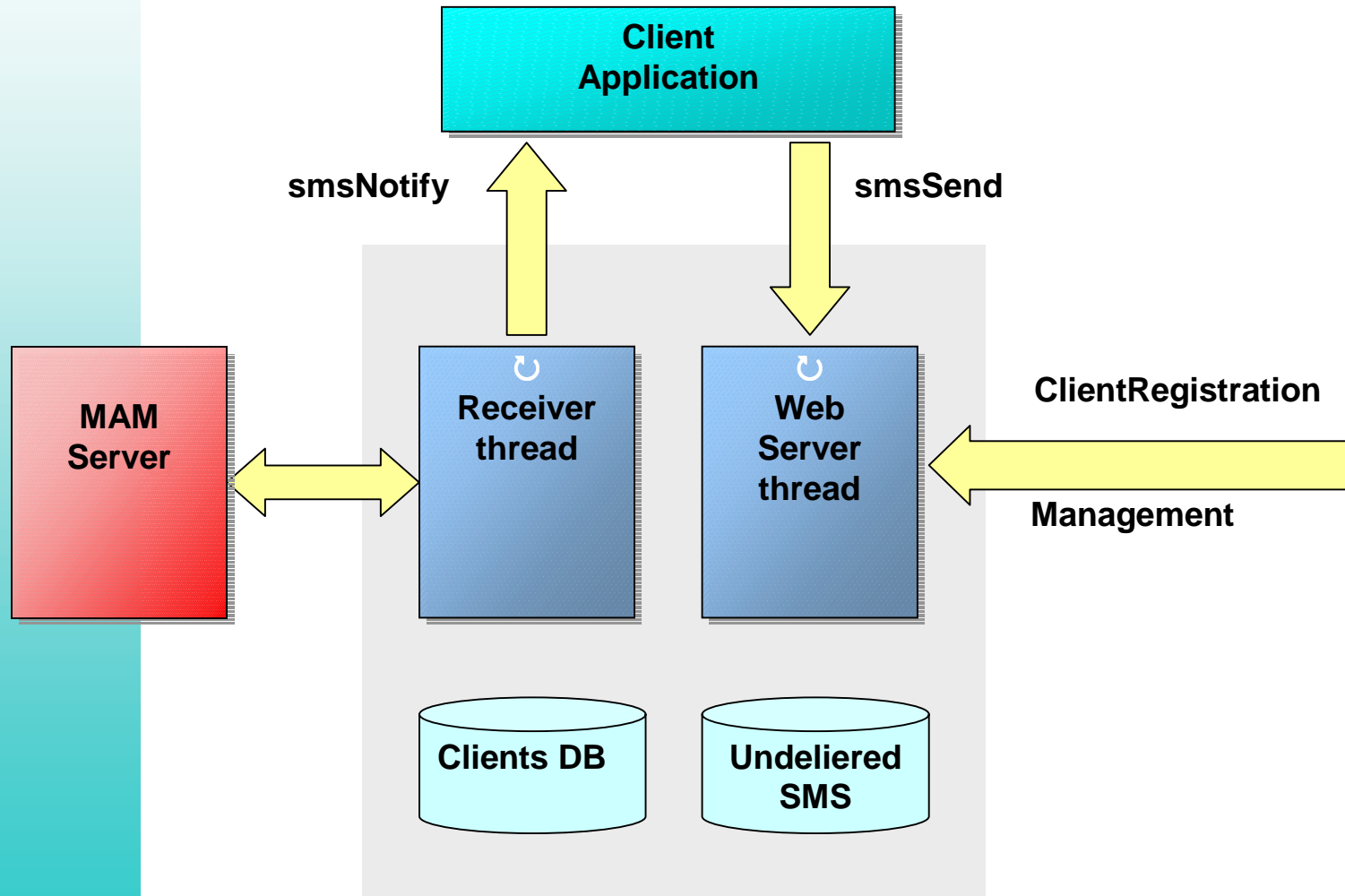
Client application



Service application



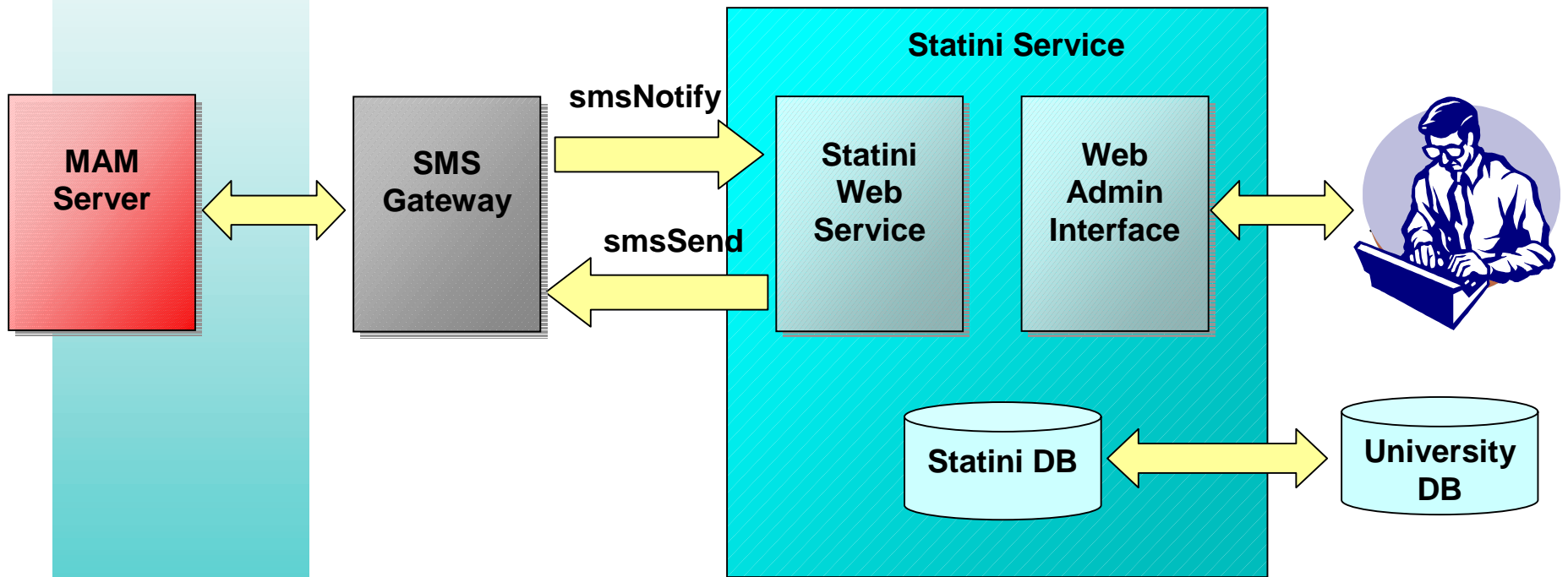
SMS Gateway Architecture



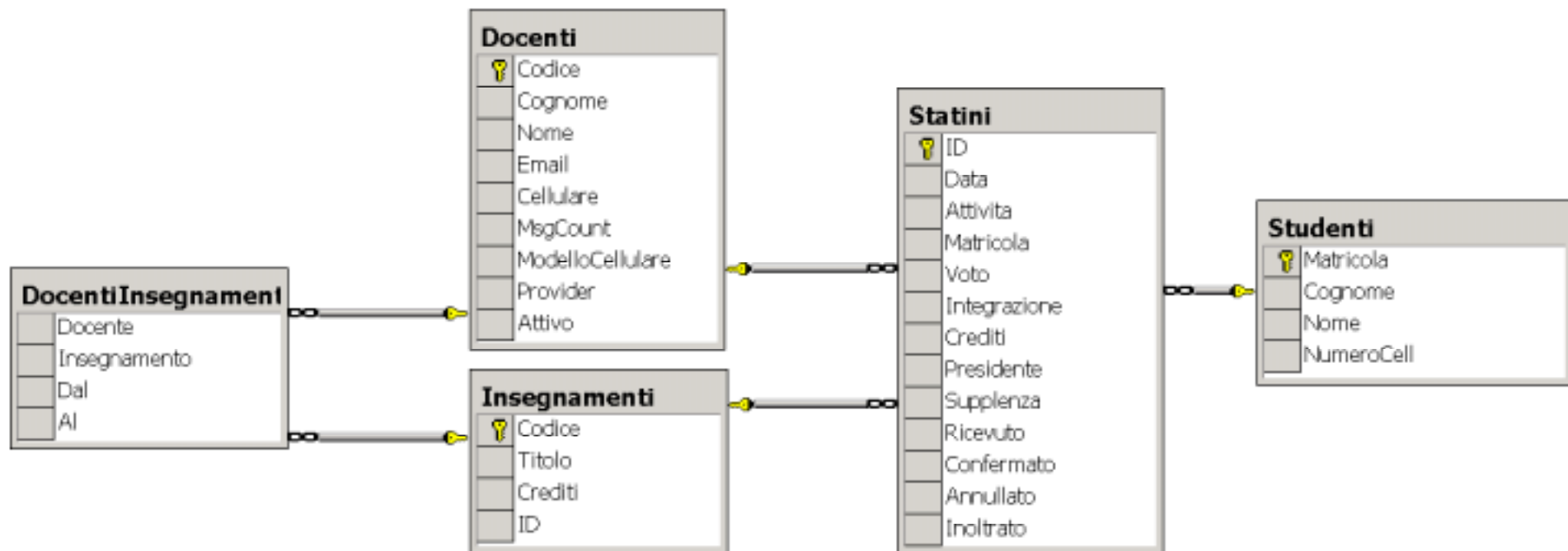
Application: Exam registration

- **Paper forms, scanned with OCR**
- **Issue: 200.000 exams each year**
- **20.000 errors**
- **Average delay: 3 months**

Exam Registration Service



Database schema



DB Object Interface

- **Generated using Raptier, a template-based code generator**
- **Two classes for each table**
 - **Docente**
 - **DocentiCollection**

DB access example

```
// get collection of all teachers
docenti = new DocentiCollection(db);
// get Docente with given codice
Docente d = docenti[codice];

// get courses taught by given teacher
cmd.CommandText = @"SELECT Insegnamenti.* FROM ...
    WHERE (DocentiInsegnamenti.Docente = " + codice +
        ")";
IDataReader rdr = cmd.ExecuteReader();
ArrayList courses = new ArrayList();
while (rdr.Read()) {
    Insegnamento course = new Insegnamento(rdr);
    courses.Add(course);
}
```

demo





Benefits

- **Cheap deployment:**
 - no special hardware
 - no software installation on clients
- **real-time feedback**
- **Immediate data validation**
- **Eliminate all paperwork**

Issues

- **Security**
 - **SMS travels on SS7 signaling network**
 - **One time passwords**
- **Legal validity**
 - **Signature by professors**
 - **Signature by students**

Conclusion

- **SMS gateway**
- **Future MMS**
- **Security enabled through embedded Java on cell phones**