Design and Implementation of a Message-Based Regional Telemedicine System to Achieve High Availability and Scalability

Xiaoyang Ren, MD, Zhenbo Wang, MD, Yajie Wu, MD, Yushen Li, MD, Mingxia Chen, MD, Yunkai Zhai, PhD and Yuhong Li, MD

+A bit of Encryption

2018-2019 Seminar medical informatics (INFOMSMI) - Stamatis Kantiloros

Inspiration for this Project

High cost of architecture redesign and development

Small Glossary of Terms

regional telemedicine system (RTS)

regional collaborative medical service system (RCMS)

What is Telemedicine?

Telemedicine is an integration between technological and clinical applications.

It uses IT to provide medical services to individuals who are far away from the medical services provider.

What a Regional Telemedicine System should do

- Synchronization of personal information about experts.
- Cross-area teleconsultation
- Small-scale telemedicine system can join the RTS.
- Support for multiple (programming) languages.

Synchronization of info about experts

There are 2 flows for the information:

Local experts of a small-scale telemedicine systems join the RTS (uplink synchronization).

RTS sharing their information with other small-scale telemedicine systems (downlink synchronization).

Cross-area teleconsultation

An applicant should be able to read the information about the registered experts and request a candidate of his choice for teleconsultation.

So the applicant sends his medical information to his chosen expert and gets back a diagnosis from them.

Small Scale telemedicine system joins RTS

First the local small-scale telemedicine system uploads the information about local experts to the RTS.

Then, RTS executes a batching process to store and share information with other systems.

Support for multiple languages

Small-scale telemedicine systems have been developed from different suppliers at different dates using different tools.

A proper RTS implementation should support a variety of programming languages so that there is no need for the existing systems to be totally reconstructed. Software development kits along with relevant documents should be provided by the creators of the RTS.

Persistence

All uplink and downlink messages should be in PERSISTENT state.

In general, in Computer Science persistence refers to a characteristic of a state that outlives the process that created it.

Dead Letter Queue

Store messages that occur in faulty scenarios like:

Message sent to a queue that does not exist

Queue length exceeded its limits

Message length exceeded its limits

etc.

What is (Apache) ZooKeeper?

- ZooKeeper aims at distilling the essence of these different services into a very simple interface to a centralized coordination service.
- The service itself is distributed and highly reliable. Consensus, group management, and presence protocols will be implemented by the service so that the applications do not need to implement them on their own.
- Simply: a message-based service medium

What is (Apache) ActiveMQ?

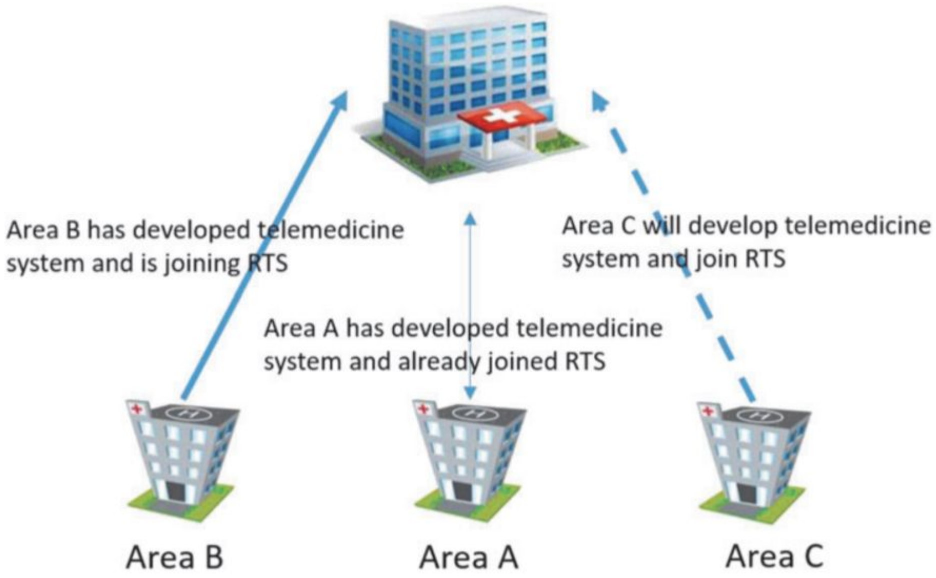
- Apache ActiveMQ is an open source message broker written in Java together with a full Java Message Service (JMS) client. It provides "Enterprise Features" (fostering the communication from more than one client or server).
- Supported clients include Java (via JMS 1.1) as well as several other "cross language" clients.The communication is managed with features such as computer clustering and ability to use any database as a JMS persistence provider besides virtual memory, cache, and journal persistency.

What is Apache?!

Apache HTTP Server, or just *Apache*,

is a free and open-source-cross-platform web server software developed by the Apache Software Foundation.

Goal of this Research Implementation


Area B has developed telemedicine system and is joining RTS

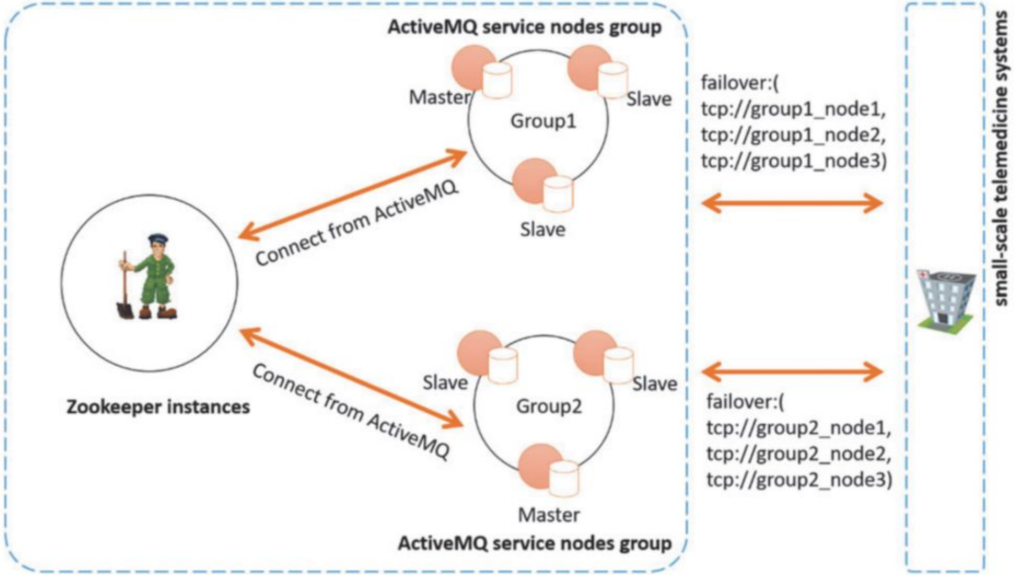
Area C will develop telemedicine system and join RTS

Area A has developed telemedicine system and already joined RTS

Goal of this Research Implementation

Centralized management of established telemedicine systems without restructuring their framework.

Hardware design diagram



Results of the Project

6 months to develop

8 small-scale telemedicine systems from 8 different cities were integrated (originally developed in Java or C#)

After 13 months of the official launch the system was stabilized

Improvements in service availability(amount of time that the system is not functional) and great potential for scalability

Security?

But what about the security level of the database?!

Sadly the authors do not disclose any information about this.

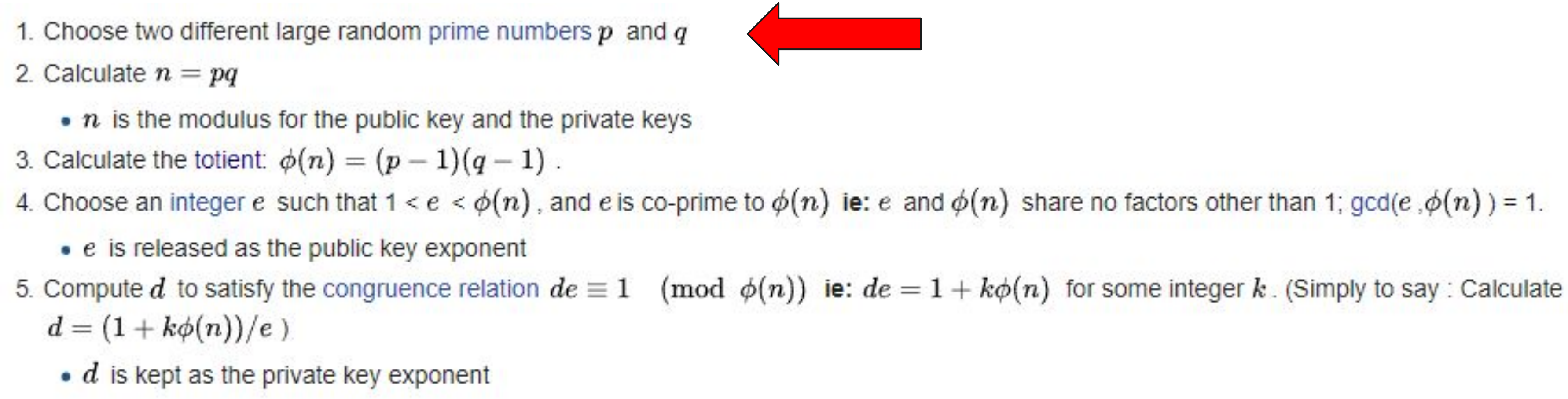
But the slides that follow discuss some general topics about encryption and how it is implemented!

RSA Algorithm

RSA was first publicly described in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman of the Massachusetts Institute of Technology.

RSA involves a public key and a private key. Messages are encrypted using the public key and decrypted using the private key. The generation of the two keys depend on prime numbers.

RSA Algorithm



1. Choose two different large random prime numbers p and q
2. Calculate $n = pq$
   - n is the modulus for the public key and the private keys
3. Calculate the totient: $\phi(n) = (p-1)(q-1)$.
4. Choose an integer e such that 1 < e < $\phi(n)$, and e is co-prime to $\phi(n)$ **ie:** e and $\phi(n)$ share no factors other than 1; gcd(e ,$\phi(n)$) = 1.
   - e is released as the public key exponent
5. Compute d to satisfy the congruence relation $de \equiv 1 \pmod{\phi(n)}$ **ie:** $de = 1 + k\phi(n)$ for some integer k . (Simply to say : Calculate $d = (1 + k\phi(n))/e$)
   - d is kept as the private key exponent

RSA Algorithm

So we see everything depends on 2 large prime numbers. But why?

There is no known efficient and deterministic way to figure out if a large number is prime or not!

There is effective statistical testing however along with bounds for their distance from each other.

What is Encryption?

Encrypting a message, or in general any data, entails applying to it a (mathematical) transformation

CIPHER: An algorithm that converts Plain Text to CipherText.

First notable historical algorithm: Caesar’s Cipher

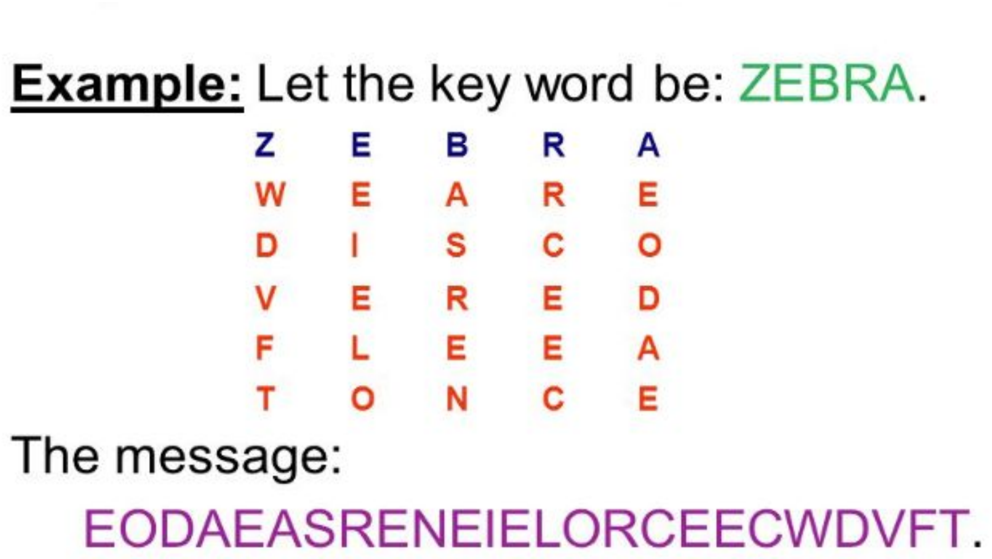
Caesar used a simple rule: Just shift all letters to the left or to the right by a set number of positions. The exact shift parameter here is the *key*.

Caesar’s Cipher

```
Plain:    ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher:   XYZABCDEFGHIJKLMNOPQRSTUVW
```

```
Plaintext:  THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Ciphertext: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD
```

Columnar Transposition Cipher



Example: Let the key word be: ZEBRA.

Z	E	B	R	A
W	E	A	R	E
D	I	S	C	O
V	E	R	E	D
F	L	E	E	A
T	O	N	C	E

The message:

EODAEASRENEIELORCEECWDVFT.

Security (Personal Opinion)

Usability

Vs

Security Level

Requires some compromises!

Question 1

What was the inspiration for developing the RTS?

Question 2

How many small-scale telemedicine systems were integrated at first launch?

Question 3 (Bonus!)

What is a general definition of Encryption ?