# Developer to CTO

CS 7002

Romain Untereiner

# Developer to CTO?

Discover the globality of IT

Be able to structure efficient and reliable IT projects

Good coding skills are not sufficient to be a good “tech”

*“Nothing is to be feared, it is only to be understood” - Marie Curie*

# Why me?

I am 28 years old and a CTO, a telecom engineer and a “touche à tout”.

Coder since too many years, did a bit of everything.

Now, more specialised in architecture, infrastructure.

2 years as telco engineer / architect @Legos, Paris
2 years as CTO @OneRagtime, Paris / Barcelona

I love to distill  complex things into simple terms

I think technology should be useful, like progress should too

# Kamina

Kamina brings a CTO as a service for SMBs

- Development, hosting and governance of your web services
- Tech advisory for companies
- Teaching

# Cantoo

The simplest and fairest access to telecom ressources

It is a **CPaaS** with **qualitative** and **sovereign services** at a **right price**.

We enable **small or foreign telecom actors** to **beneficiate** from **modern operator services**.

- Number activation, hosting and portability
- Voice call
- Messaging

# Our 3-weeks path

1. Explore the ICT ecosystem (week 1)
2. Embrace business & social IT-related aspects (week 2)
3. Concretely drive your technology (week 2)
4. Present your technology (week 3)

# The module orchestration

- **3 weeks**, 3h session per day
- **interaction of course**
- **daily assignments**, given and explained at beginning of each session
- **mid-module & final assignments**, put into action your learnings
- **project presentations**
- **grading**: (50%) Final Project (30%) Participation (20%) Homework

# The final project

**2 options**

- give structure and clarity to your capstone project

- starting up a new project design

What do you prefer?

# What do you know about yourself ?

Which geek are you?

Do you like what you do?

Do you want to something in special to do?

# What do you know about IT ?

What is IT?

What is the thing that makes you like IT?

# What do you would like to know about IT ?

- Architecture aspects
- Governance aspects
- Coding aspects
- Infrastructure aspects
- Social aspects
- ...

# Introduction to ICT

Session 1

# ICT

Information and Communications Technology (or Technologies)

“A **technology** that **treats** or **move information**“

# Everything is information

*Informare* is to “give a sens”

Information can be thought of as the resolution of uncertainty

Information is entropy

It is knowledge than can have an effect

The difference between data and information?

# Everything is communication

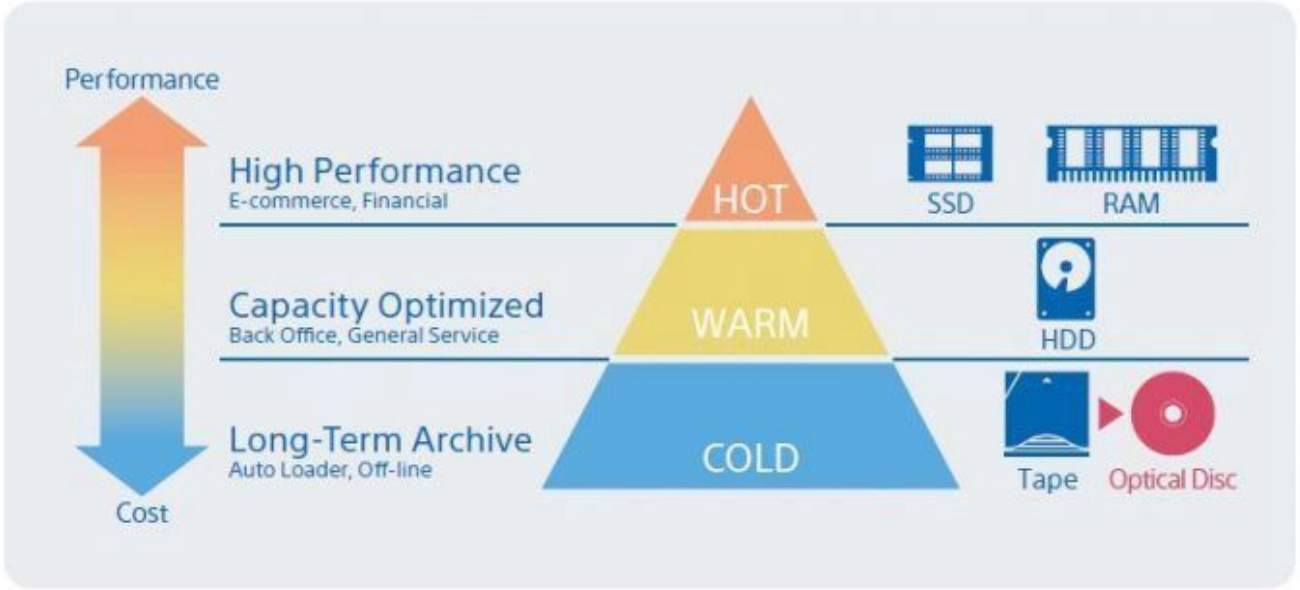
*Communicare* is to “put is common”

Communication is relation

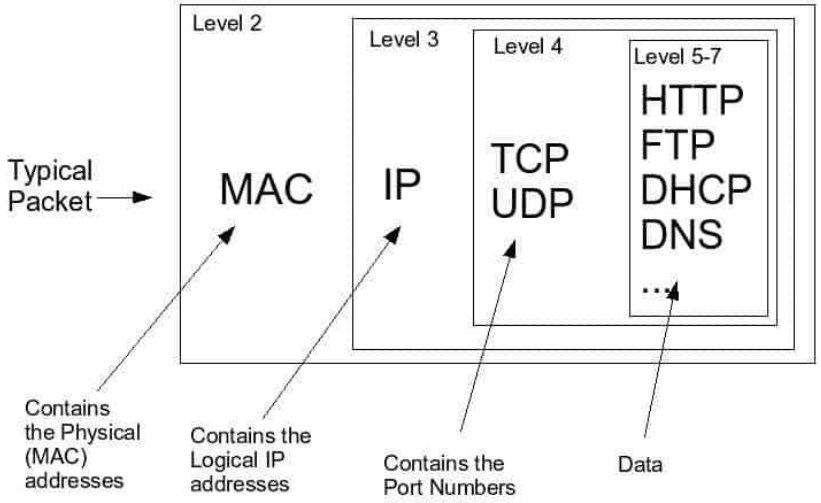
Communication has to deal with time

The difference between communication and information?

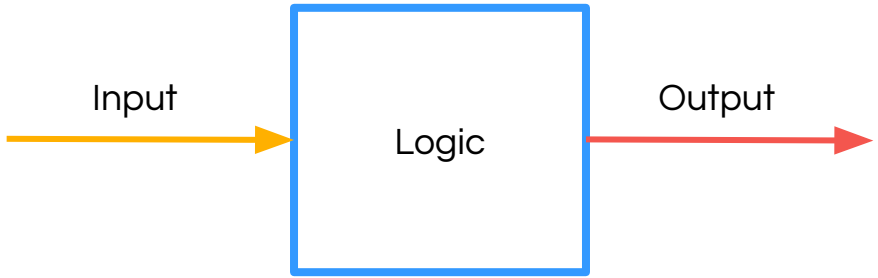
# Stationary information: Data



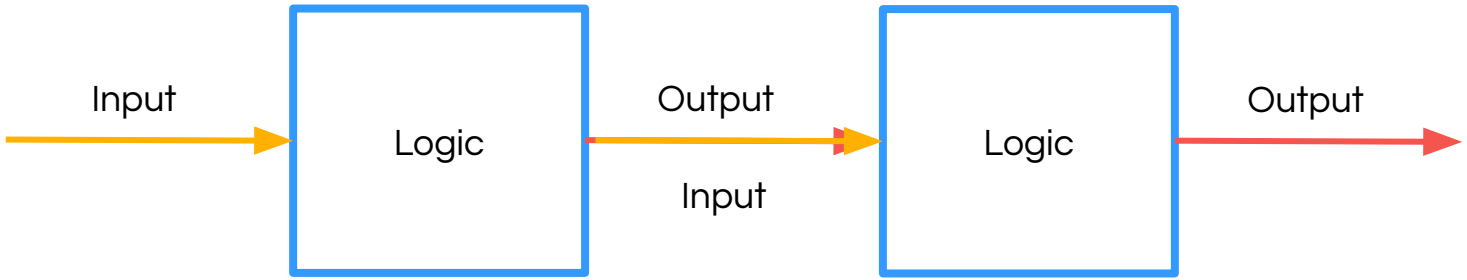
# Moving information: Packets



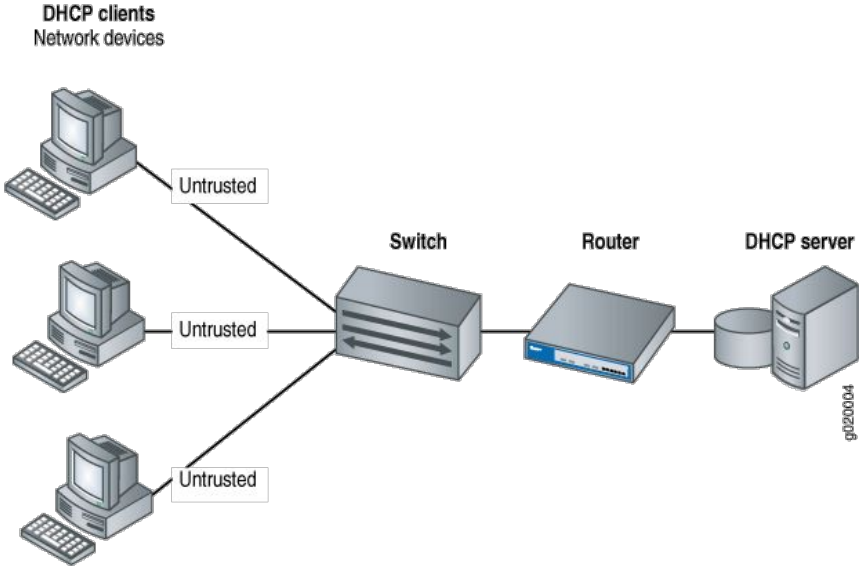
# One scheme to know them all



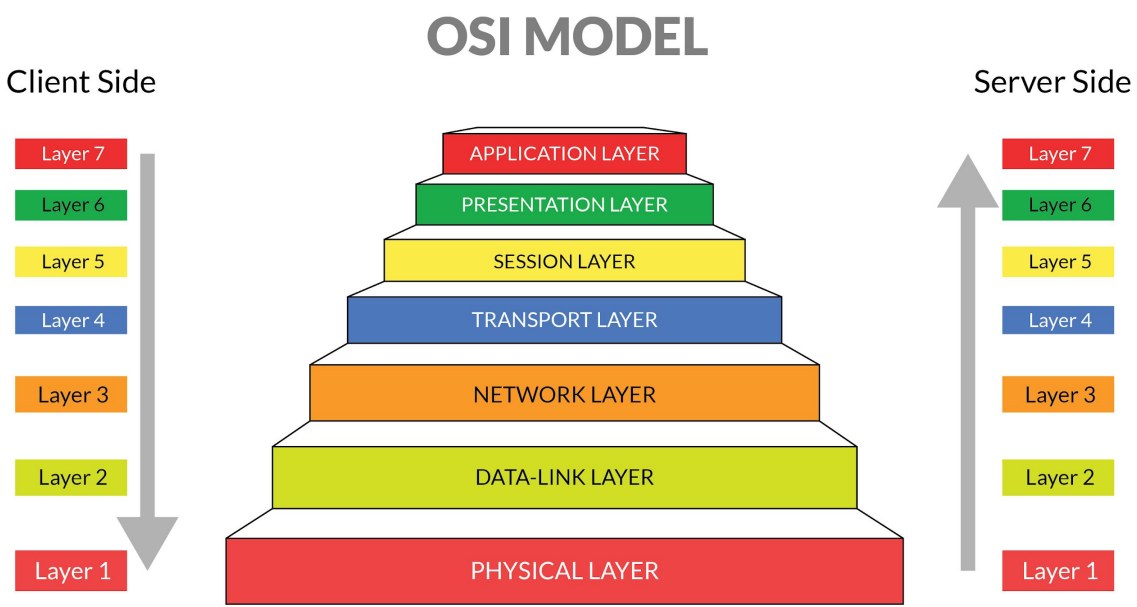
# One scheme to know them all



# Some networking



# OSI layers


OSI MODEL
Client Side
Server Side
Layer 7
Layer 6
Layer 5
Layer 4
Layer 3
Layer 2
Layer 1
APPLICATION LAYER
PRESENTATION LAYER
SESSION LAYER
TRANSPORT LAYER
NETWORK LAYER
DATA-LINK LAYER
PHYSICAL LAYER
Layer 7
Layer 6
Layer 5
Layer 4
Layer 3
Layer 2
Layer 1

# OSI layers

## Comparison to other networking suites [ edit ]

The table below presents a list of OSI layers, the original OSI protocols, and some *approximate* modern matches. It is very important to note that this correspondance is rough: the OSI model contains idiosyncracies not found in later systems such as the IP stack in modern Internet.[19]

| Layer | | OSI protocols | TCP/IP protocols | Signaling System 7[32] | AppleTalk | IPX | SNA | UMTS | Miscellaneous examples |
|---|---|---|---|---|---|---|---|---|---|
| No. | Name | | | | | | | | |
| 7 | Application | FTAM · X.400 · X.500 · DAP · ROSE · RTSE · ACSE[33] · CMIP[34] | HTTP · HTTPS | INAP · MAP · TCAP · ISUP · TUP | AFP · ZIP · RTMP · NBP | SAP | APPC | | HL7 · Modbus · WebSocket |
| 6 | Presentation | ISO/IEC 8823 · X.226 · ISO/IEC 9576-1 · X.236 | MIME · SSL/TLS · XDR | | AFP | | | | TDI · ASCII · EBCDIC · MIDI · MPEG |
| 5 | Session | ISO/IEC 8327 · X.225 · ISO/IEC 9548-1 · X.235 | Sockets (session establishment in TCP / RTP / PPTP) | | ASP · ADSP · PAP | NWLink | DLC? | | Named pipes · NetBIOS · SAP · RPC · SOCKS |
| 4 | Transport | ISO/IEC 8073 · TP0 · TP1 · TP2 · TP3 · TP4 (X.224) · ISO/IEC 8602 · X.234 | TCP · UDP · SCTP · DCCP | | DDP | SPX | | | NBF |
| 3 | Network | ISO/IEC 8208 · X.25 (PLP) · ISO/IEC 8878 · X.223 · ISO/IEC 8473-1 · CLNP X.233 · ISO/IEC 10589 · IS-IS | IP · IPsec · ICMP · IGMP · OSPF · RIP | SCCP · MTP | ATP (TokenTalk / EtherTalk) | IPX | | RRC / BMC | NBF · Q.931 |
| 2 | Data link | ISO/IEC 7666 · X.25 (LAPB) · Token Bus · X.222 · ISO/IEC 8802-2 · LLC (type 1 / 2)[35] | PPP · SBTV · SLIP | MTP · Q.710 | LocalTalk · ARA · PPP | IEEE 802.3 framing Ethernet II framing | SDLC | PDCP[36] · LLC · MAC | ARP · NDP (Neighbor Discovery Protocol) · ARQ · ATM · Bit stuffing · CDP · DOCSIS · FDDI · FDP · Fibre Channel · Frame Relay · HDP · HDLC · IEEE 802.3 (Ethernet) MAC · IEEE 802.11 (Wi-Fi) MAC · IEEE 802.1Q (VLAN) · ISL · ITU-T G.hn DLL · Linux interface bonding · PPP · Q.921 · Token Ring · NDP (Nortel Discovery Protocol) · IS-IS |
| 1 | Physical | X.25 (X.21bis · EIA/TIA-232 · EIA/TIA-449 · EIA-530 · G.703)[35] | | MTP · Q.710 | RS-232 · RS-422 · PhoneNet | | Twinax | UMTS air interfaces | RS-232 · RJ45 (8P8C) · V.35 · V.34 · I.430 · I.431 · T1 · E1 · 802.3 PHY (10BASE-T · 100BASE-TX · 1000BASE-T) · POTS · SONET · SDH · DSL · 802.11 PHY · ITU-T G.hn PHY · DOCSIS · DWDM · OTN |

# Some important basics

Client and server vs binded mode

Connected vs non-connected mode

Stateless vs stateful

SPOF

Redundancy? Availability? Reliability?

KPI

# Do you see some IT around?

**IT is everywhere** - in your personal and professional life

We always **use** IT

But you will often be there to **build** IT, **using** other IT elements

# “Coding” can be a part of IT

“Coding” is designing, developing, delivering and supporting a electronic / digital service or product for someone that is having value from it.

I call it “to build tech”.

The core of your purpose you are building
It has a delivery / service level agreement with a customer.

**Making or using** // **Build or Support function**

# Build blocks together

- Interface(s)
- Software piece(s)
- Data schema(s)
- Data source(s)
- Running instance(s)
- Monitoring(s) & Alerting(s)
- Networking(s)

# Functional elements

- Project documents

- Visual assets

- Domain name

- Interface: GUI, API, command, product

- Tools: e-mail, tasks management, analytics, productivity tools

- Logic

- Aggregate

- Data storage

# Technical elements

- Code source
- Data model
- Data source(s)
- Running instance(s)

# See the world as functions

- Human has cold and warm information storage, and communicates mostly with its voice interface.

- Human to human communication OSI layers would be:

    - Sound waves
    - Language
    - Form
    - Substance
    - Serialization = understanding

- We are aggregating external world information with our sensors and process it in background to make knowledge (or not), with SLAs (or not)

# Session learnings

- ICT is: ...
- Everything is information:  input -> logic -> output
- Networking & OSI
- What makes “tech builders” different
- Act local, see global

# Next session

**Build tech**

- What is to build tech?
- How is formalized IT?
- How is formalized Tech?

# For the next session

- Fill the “Introductory form”

For me to know you and your thinkings

- Fill the “IT Skills / Wills” matrix
- Do the MBTI test
- Choose what to do for within 2 weeks