THE DEVELOPER'S CONFERENCE

"Operando com Go"

Digital Surgery and Integration with Medical Devices

Leonardo Zamariola

2021-03

About me

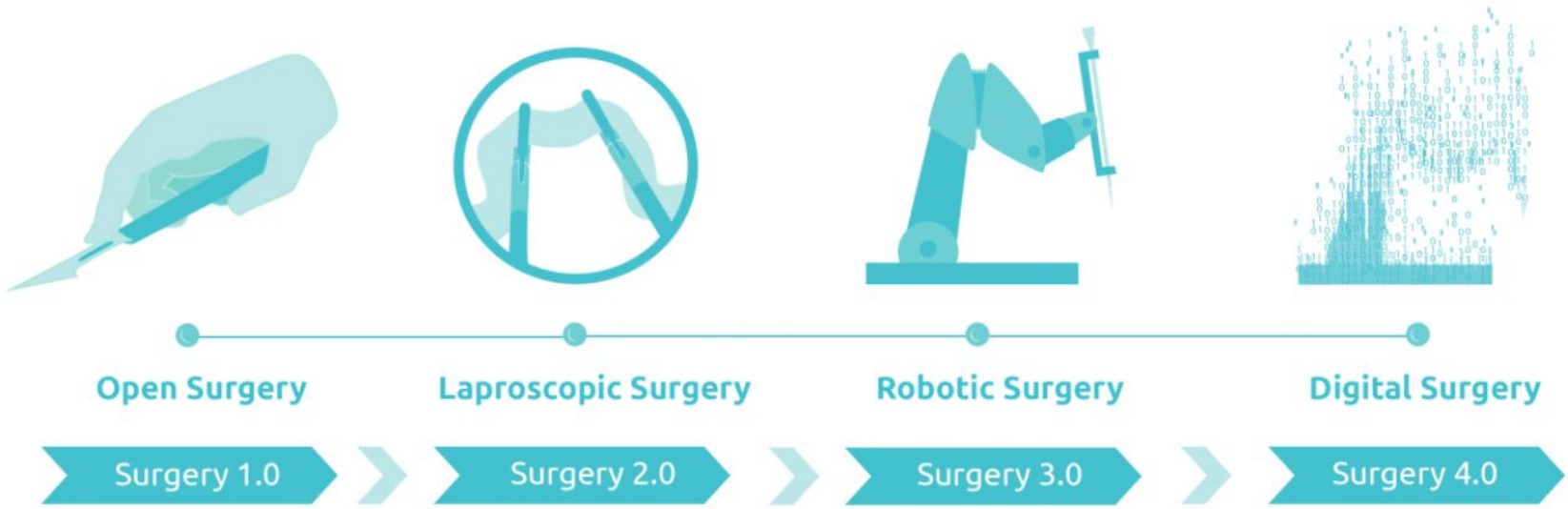
Staff Software Engineer @ CI&T

- C, C#, Java, Kotlin, Go, Python
- Oracle Certified Associate
- Google Professional Cloud Architect

- E-commerce
- Mobile Applications / Backend
- System Integrations
- Digital Surgery / Medical Devices

www.zamariola.com.br

What is digital surgery



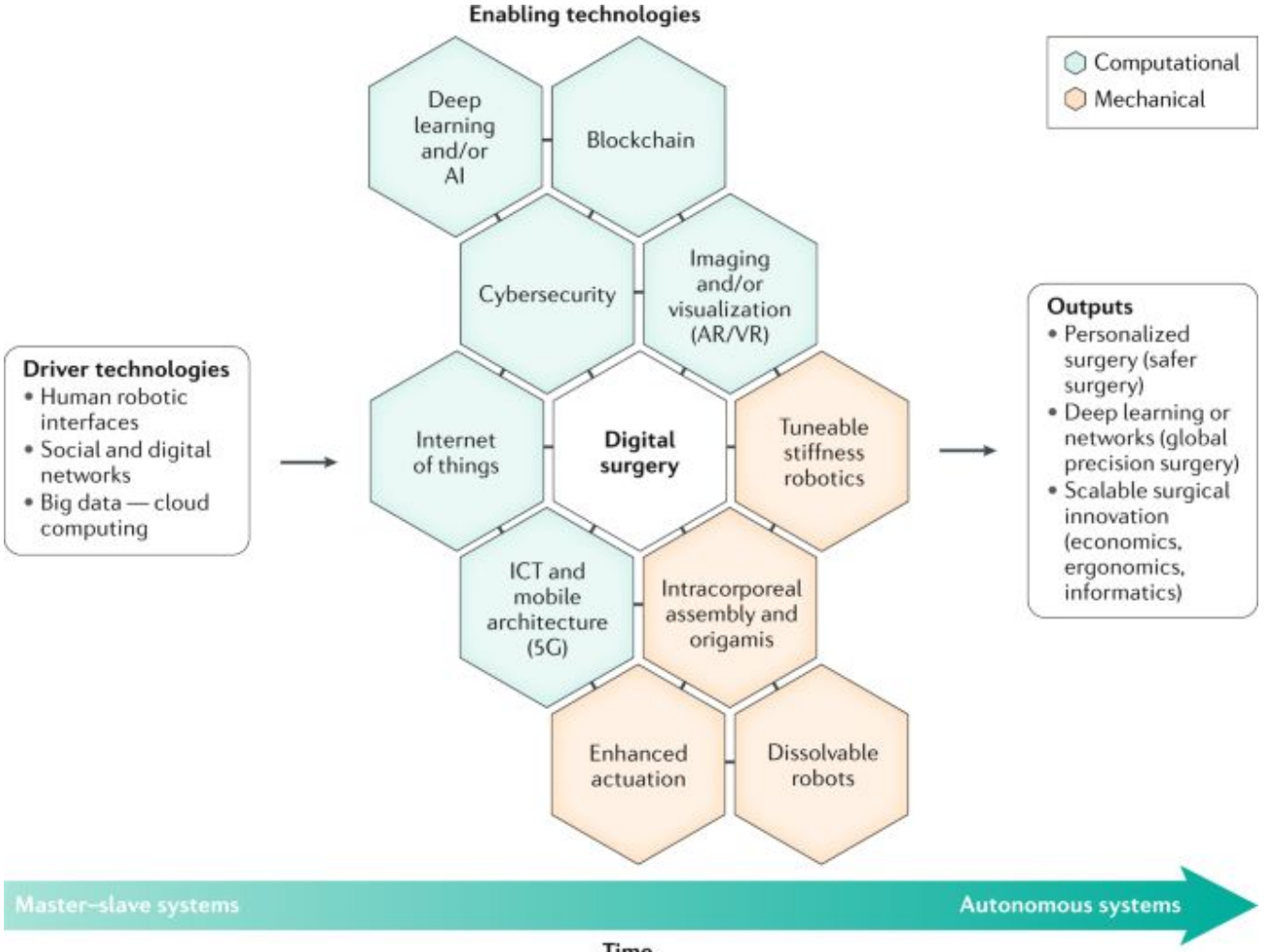
https://digitalsurgery.com/2019/05/21/digital-surgery-seeds-in-fertile-soil/

What is digital surgery

"The arrival of accurate and actionable **data can facilitate the realization of value-based care in surgery**, allowing for a better understanding of the patterns of care delivery against the patterns of costs and outcomes"

"Robots will become smaller [...] Without a doubt, these advancements will make robotic surgery relevant to a wider range of procedures. However, it is only at the intersection of **AI and data that robotic surgery can truly address the elephant in the room: surgical variation**. Surgery, although a technically-demanding profession, is largely dependent on the decision-making capabilities of the surgeon."

https://digitalsurgery.com/2019/05/21/digital-surgery-seeds-in-fertile-soil/


Enabling technologies
Computational
Mechanical
Deep learning and/or AI
Blockchain
Cybersecurity
Imaging and/or visualization (AR/VR)
Internet of things
Digital surgery
Tuneable stiffness robotics
ICT and mobile architecture (5G)
Intracorporeal assembly and origamis
Enhanced actuation
Dissolvable robots
Driver technologies
• Human robotic interfaces
• Social and digital networks
• Big data — cloud computing
Outputs
• Personalized surgery (safer surgery)
• Deep learning or networks (global precision surgery)
• Scalable surgical innovation (economics, ergonomics, informatics)
Master–slave systems
Autonomous systems
Time

Input

openEMR

FHIR®

DICOM™
Digital Imaging and Communications in Medicine

HL7® International

Process

Use

The average age of all fixed assets in the U.S. — including hospitals and medical devices — hit **22.8 years** in 2015, the oldest number reported since 1925, according to figures from the U.S. Bureau of Economic Analysis.

This rise in the fixed asset age index stems from the unwillingness, or incapability, of companies and governments to spend money on updated equipment, facilities and products, reports *Bloomberg.*

Here are three things to know.

1. In the past two years, the **average age of medical equipment and tools reached 4.7 years** — the oldest since 1945.

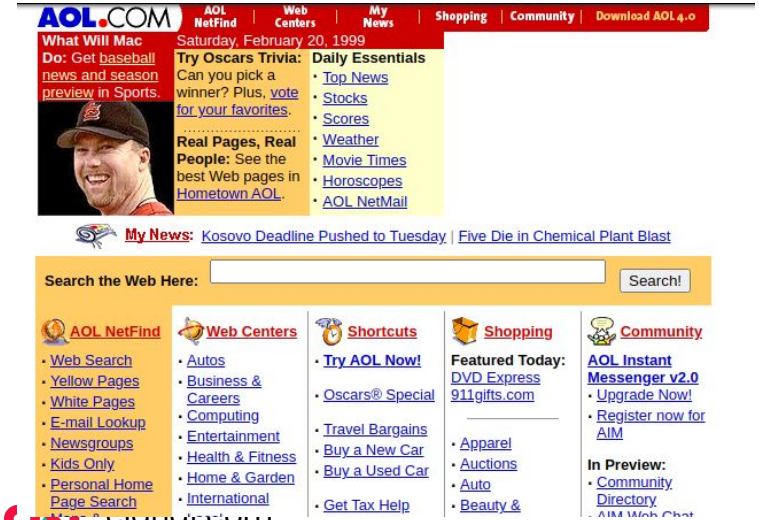
2. Healthcare facility structures demonstrated an average age of 20.7 years, the highest figure since 1949, according to the report.

3. The BEA pinpointed industrial equipment to be 10 years old, on average. In 2010 and 2011, the equipment was an average of 10.2 years old, the oldest age recorded since 1939.

https://www.beckershospitalreview.com/supply-chain/medical-equipment-hits-oldest-average-age-since-1945.html

Web 22y ago


Google!
BETA
Search the web using Google!
Google Search
I'm feeling lucky
Special Searches
Stanford Search
Linux Search
Why use Google!
Press about Google!
Help!
Company Info
Jobs at Google
Google! Logos
Making Google! the Default
Get Google!
updates monthly:
your e-mail
Subscribe
Archive
Copyright ©1999 Google Inc.


AOL.COM
AOL NetFind
Web Centers
My News
Shopping
Community
Download AOL 4.0
Saturday, February 20, 1999
What Will Mac Do: Get baseball news and season preview in Sports.
Try Oscars Trivia: Can you pick a winner? Plus, vote for your favorites.
Real Pages, Real People: See the best Web pages in Hometown AOL.
Daily Essentials
Top News
Stocks
Scores
Weather
Movie Times
Horoscopes
AOL NetMail
My News: Kosovo Deadline Pushed to Tuesday | Five Die in Chemical Plant Blast
Search the Web Here:
Search!
AOL NetFind
Web Search
Yellow Pages
White Pages
E-mail Lookup
Newsgroups
Kids Only
Personal Home Page Search
Web Centers
Autos
Business & Careers
Computing
Entertainment
Health & Fitness
Home & Garden
International
Shortcuts
Try AOL Now!
Oscars® Special
Travel Bargains
Buy a New Car
Buy a Used Car
Get Tax Help
Shopping
Featured Today:
DVD Express
911gifts.com
Apparel
Auctions
Auto
Beauty &
Community
AOL Instant Messenger v2.0
Upgrade Now!
Register now for AIM
In Preview:
Community Directory

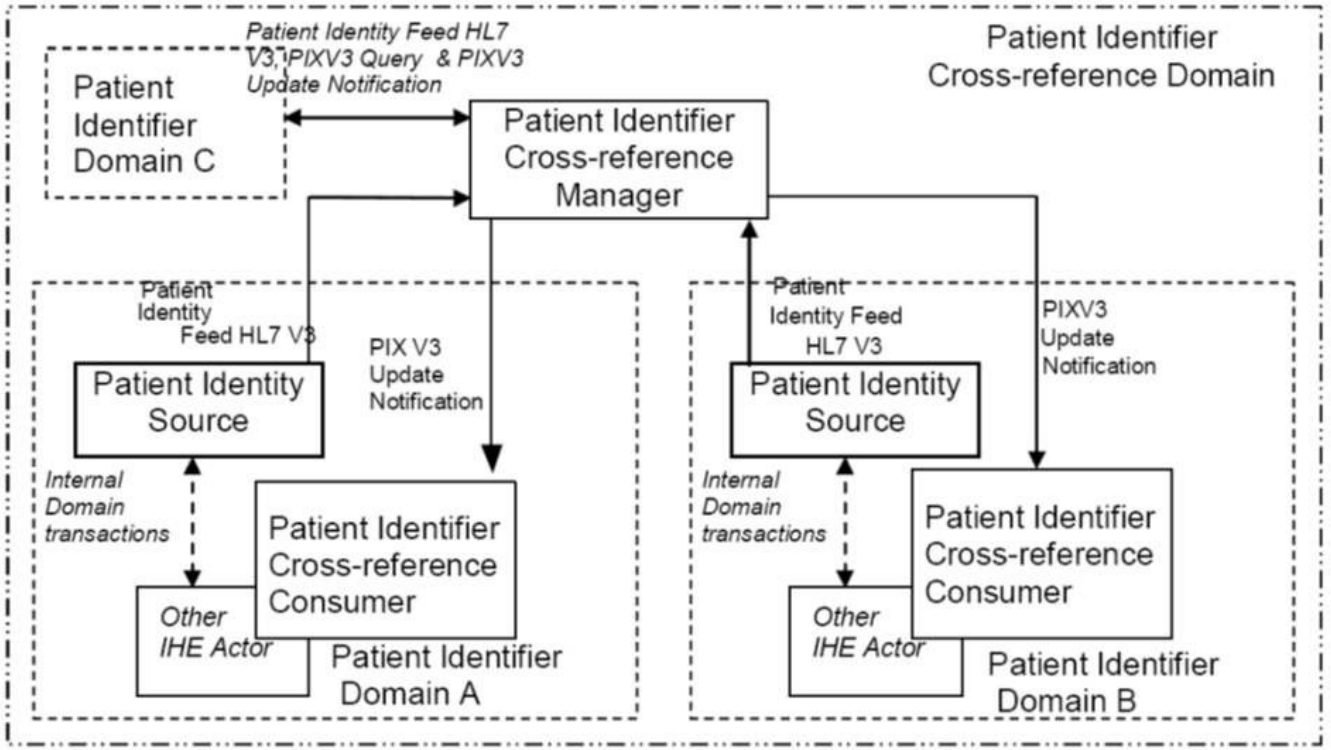

msn
Microsoft
NOVEMBER 28
SEARCH the Web
go
Home
Hotmail
Search
Shopping
Money
People & Chat
Passport sign in
Free download MSN Messenger Service
Get what you like Personalize this page
Autos
Business
Careers
Computing & Web
Entertainment
Games
Health
Holiday Fun New!
Home & Loans
Local Guides New!
MSN Update
News
Personal Finance
Radio & Video
Research & School
Sports
Travel
Women
Air Tickets
Auctions
Buy Books
Buy Music
Downloads
E-cards
Free Games
Get Online
Home Pages
Maps
Stock Quotes
More ...
Slide show: Hubble telescope images
Scenic views from space
Also today
Top 10 tax deductions
What manly men read
Save: off-season travel
Take Barbie on a trip
Make the connection
Top wireless phones
Anxiety at social events?
Send a holiday e-card
The shy man's dating guide
MESSAGE CENTER
E-mail
Hotmail Member Name:
Password:
Sign up for free e-mail
People & Chat
Build a community
Cool Communities
Emoticon Lobby Chat
Chat's new features
HELP
Only on MSN
Hungry for freebies? Find plenty here
Y2K is coming: Enter to
News
MSNBC News
Ulster Unionists adopt peace plan
Russian artillery pounds Grozny
Local Lookups
Yellow Pages
White Pages
Weather
Find a Job

Protocols

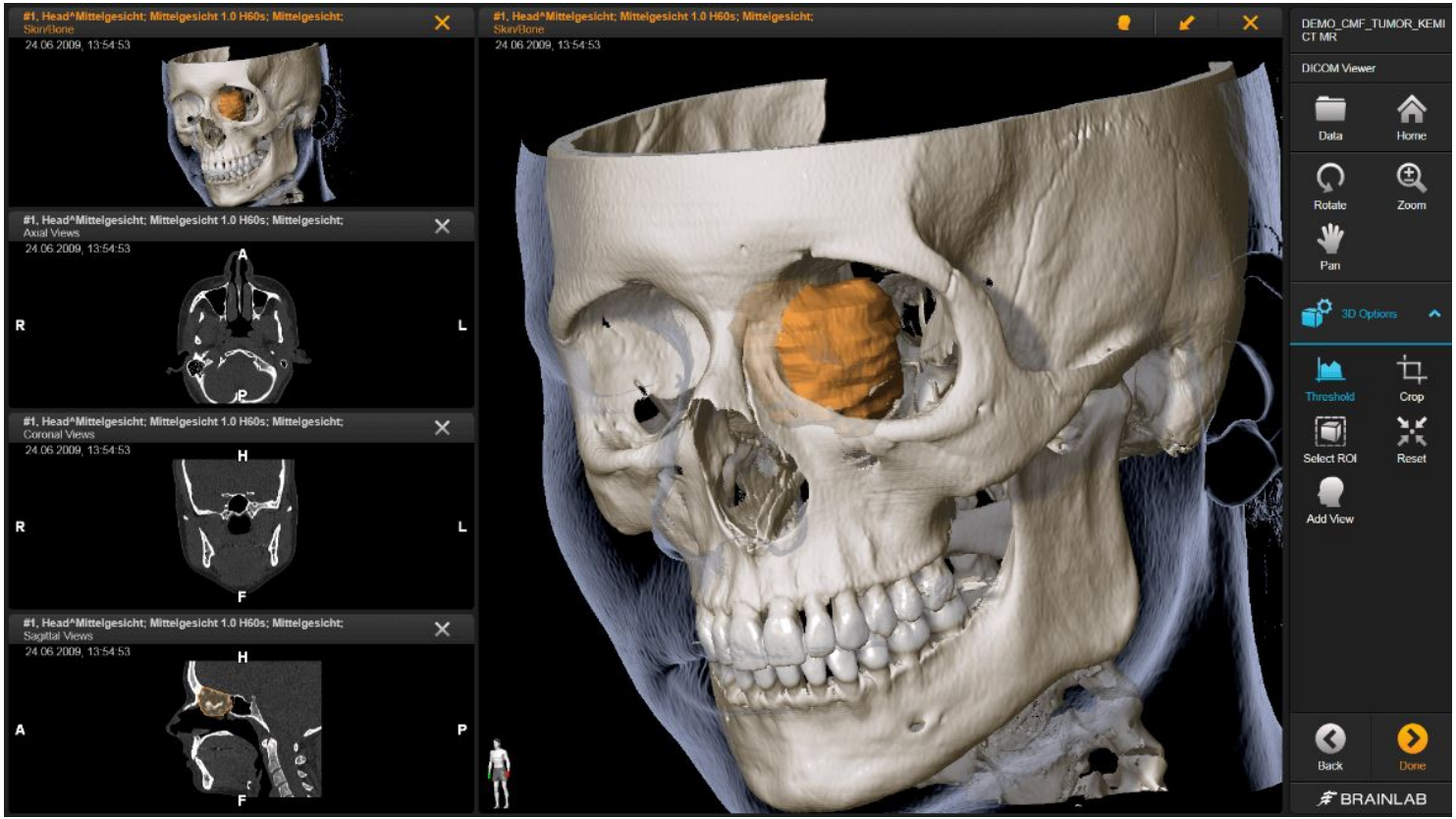
HL7 / ADT Message - 1970s

```
MSH|^~\&|MegaReg|XYZHospC|SuperOE|XYZImgCtr|20060529090131-0500||ADT^A01^ADT_A01|0105
EVN||200605290901||||200605290900
PID|||56782445^^^UAReg^PI||KLEINSAMPLE^BARRY^Q^JR||19620910|M||2028-9^^HL70005^RA9911
PV1||I|W^389^1^UABH^^^^3||||12345^MORGAN^REX^J^^^MD^0010^UAMC^L||67890^GRAINGER^LUCY^
OBX|1|NM|^Body Height||1.80|m^Meter^ISO+|||||F
OBX|2|NM|^Body Weight||79|kg^Kilogram^ISO+|||||F
AL1|1||^ASPIRIN
DG1|1||786.50^CHEST PAIN, UNSPECIFIED^I9|||A
```

HL7 / FHIR - 1987



DICOM - 1990

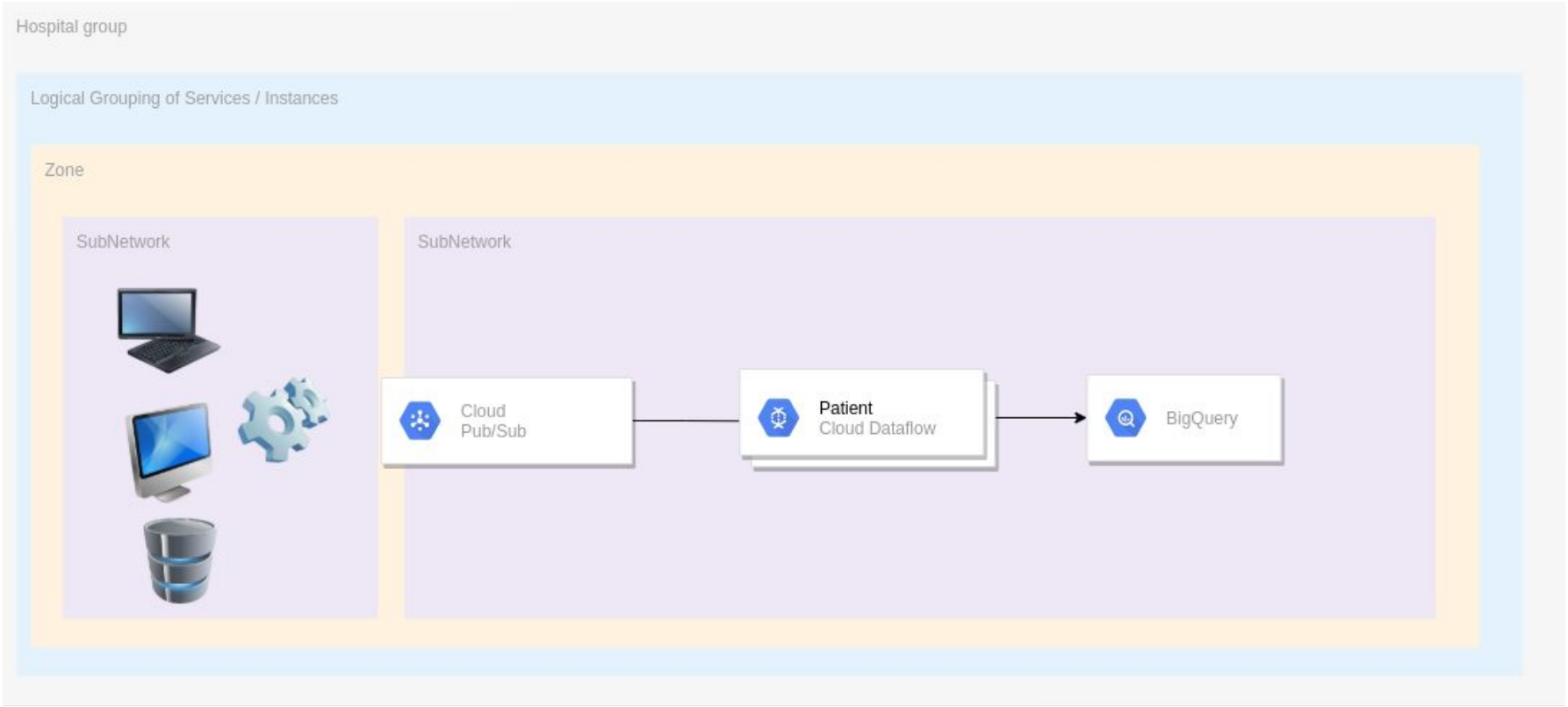

#1, Head^Mittelgesicht; Mittelgesicht 1.0 H60s; Mittelgesicht;
Skin/Bone
24.06.2009, 13:54:53
#1, Head^Mittelgesicht; Mittelgesicht 1.0 H60s; Mittelgesicht;
Axial Views
24.06.2009, 13:54:53
A
R
L
P
#1, Head^Mittelgesicht; Mittelgesicht 1.0 H60s; Mittelgesicht;
Coronal Views
24.06.2009, 13:54:53
H
R
L
F
#1, Head^Mittelgesicht; Mittelgesicht 1.0 H60s; Mittelgesicht;
Sagittal Views
24.06.2009, 13:54:53
H
A
P
F
#1, Head^Mittelgesicht; Mittelgesicht 1.0 H60s; Mittelgesicht;
Skin/Bone
24.06.2009, 13:54:53
DEMO_CMF_TUMOR_KEMI
CT MR
DICOM Viewer
Data
Home
Rotate
Zoom
Pan
3D Options
Threshold
Crop
Select ROI
Reset
Add View
Back
Done
BRAINLAB

DICOM - 1990

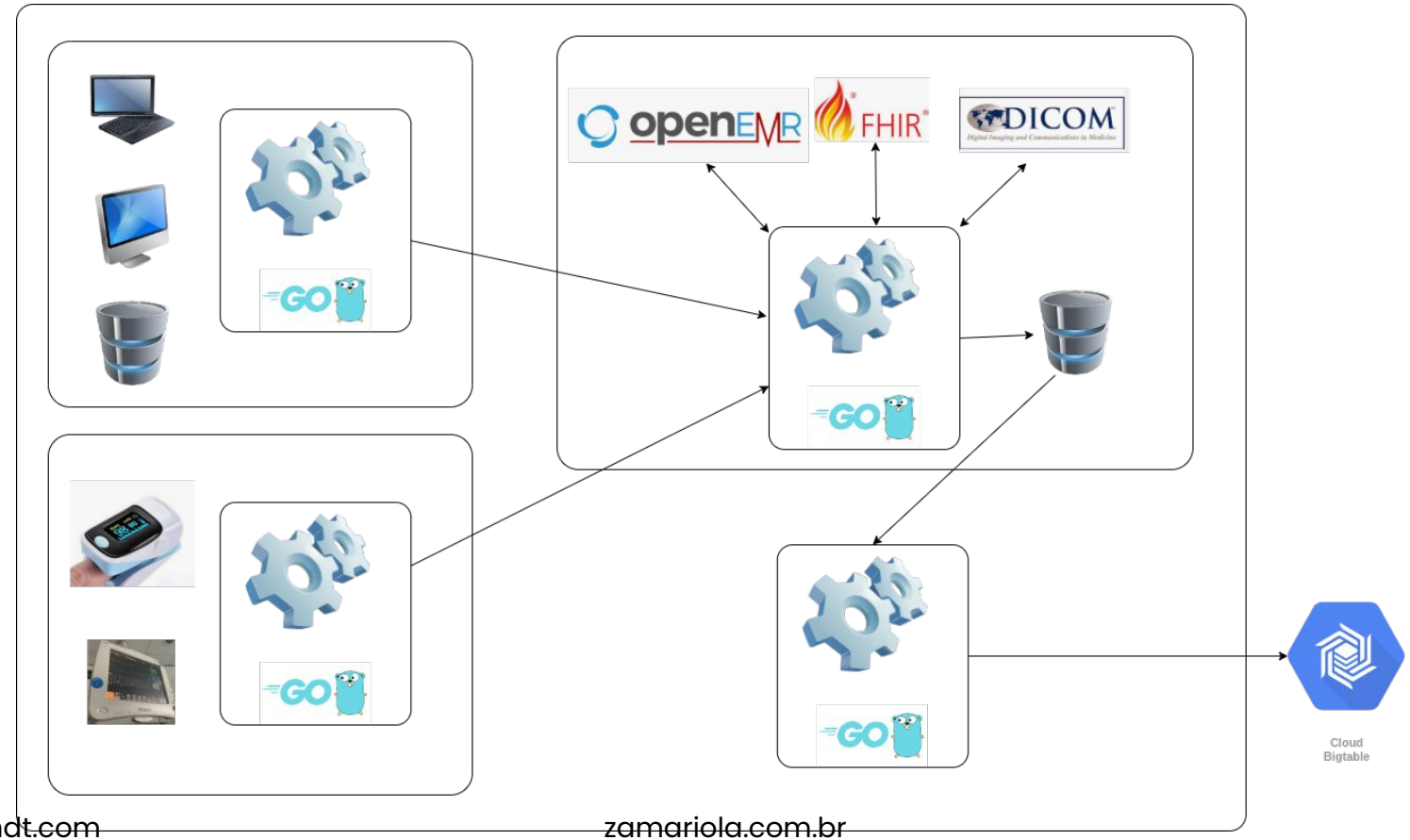
```
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000080: 4449 434d 0200 0000 554c 0400 c600 0000  DICM....UL......
00000090: 0200 0100 4f42 0000 0200 0000 0001 0200  ....OB..........
000000a0: 0200 5549 1a00 312e 322e 3834 302e 3130  ..UI..1.2.840.10
000000b0: 3030 382e 352e 312e 342e 312e 312e 3700  008.5.1.4.1.1.7.
000000c0: 0200 0300 5549 3600 312e 322e 3237 362e  ....UI6.1.2.276.
000000d0: 302e 3732 3330 3031 302e 332e 312e 342e  0.7230010.3.1.4.
000000e0: 3338 3436 3035 3336 3236 2e33 3936 382e  3846053626.3968.
000000f0: 3132 3836 3239 3939 3731 2e36 3936 0200  1286299971.696..
00000100: 1000 5549 1400 312e 322e 3834 302e 3130  ..UI..1.2.840.10
00000110: 3030 382e 312e 322e 3100 0200 1200 5549  008.1.2.1.....UI
00000120: 1c00 312e 322e 3237 362e 302e 3732 3330  ..1.2.276.0.7230
00000130: 3031 302e 332e 302e 332e 352e 3500 0200  010.3.0.3.5.5...
00000140: 1300 5348 1000 4f46 4649 535f 4443 4d54  ..SH..OFFIS_DCMT
00000150: 4b5f 3335 3520 0200 0200 5549 1a00 312e  K_355 ....UI..1.
00000160: 322e 3834 302e 3130 3030 382e 352e 312e  2.840.10008.5.1.
00000170: 342e 312e 312e 3700 0800 1600 5549 1a00  4.1.1.7.....UI..
00000180: 312e 322e 3834 302e 3130 3030 382e 352e  1.2.840.10008.5.
00000190: 312e 342e 312e 312e 3700 0800 1800 5549  1.4.1.1.7.....UI
000001a0: 3600 312e 322e 3237 362e 302e 3732 3330  6.1.2.276.0.7230
000001b0: 3031 302e 332e 312e 342e 3338 3436 3035  010.3.1.4.384605
000001c0: 3336 3236 2e33 3936 382e 3132 3836 3239  3626.3968.128629
000001d0: 3939 3731 2e36 3936 0800 2100 4441 0800  9971.696..!.DA..
000001e0: 3230 3130 3130 3035 0800 3100 544d 0600  20101005..1.TM..
000001f0: 3133 3332 3531 0800 3e10 4c4f 0600 5452  133251..>.LO..TR
```

And everything else...

Traditional approach (map-reduce)



EHR - HIPAA - GDPR - CCPA - “PMB” approach



Why Go?

- Embedded dependencies
- Single binary -> target platform
- Low resource usage - > Doctor's computer / nfs
- Networking std library -> net.ListenTCP()
- Binary manipulation
- Speed -> DICOM file up to GBs / monitoring: each device with messages every half second
- Easy to learn and to code -> build tools/ test tools

Talk is Cheap! Show me the code!

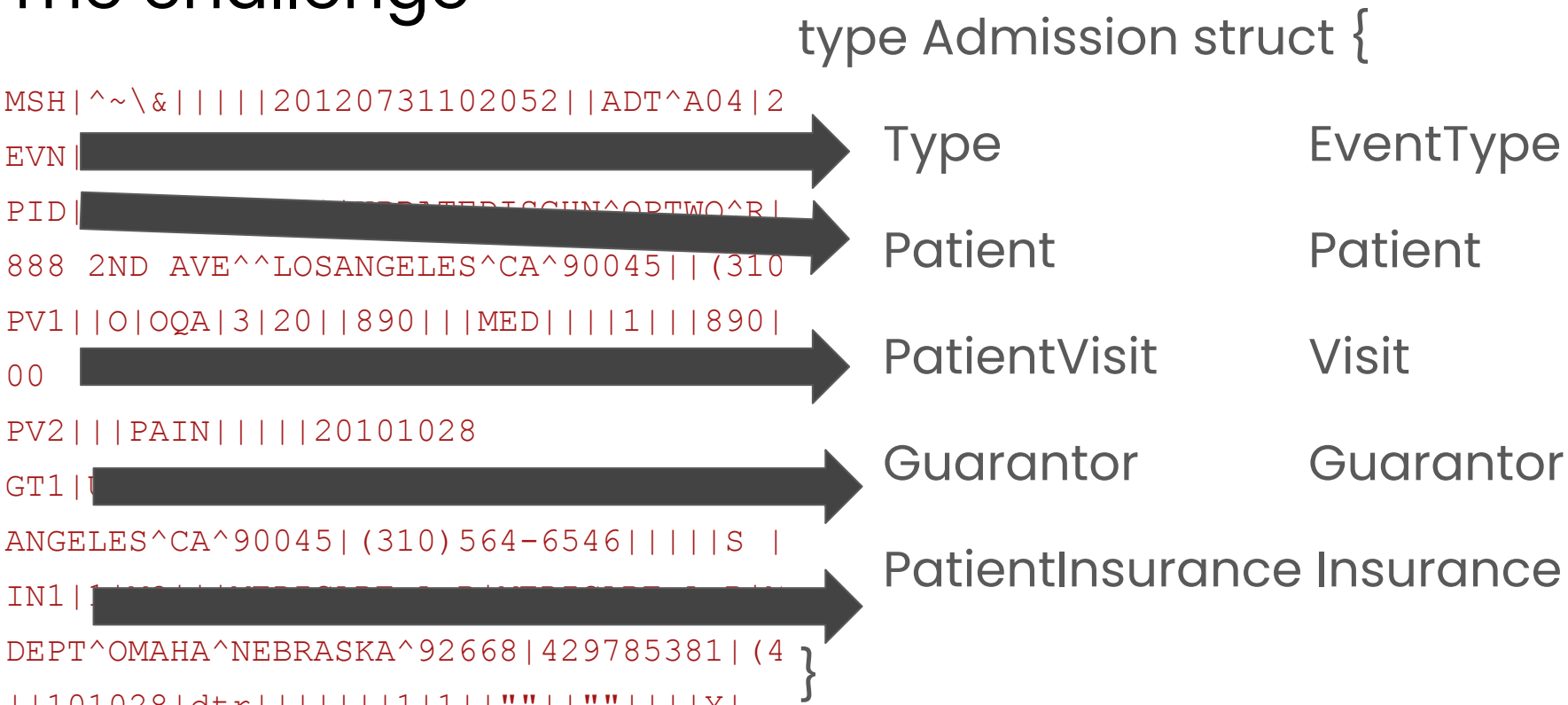
The challenge


```
MSH|^~\&|||||20120731102052||ADT^A04|207311720520001553 74|P|2.2|18
EVN|A04|20120731102052|||dtrn1
PID|||000913545||UPDATEDISCHN^OPTWO^R||19410115|M|||
888 2ND AVE^^LOSANGELES^CA^90045||(310)564-6546|||M|AT|155374|949-84-9494
PV1||O|OQA|3|20||890|||MED||||1|||890|O|||||||||||||||||||||||||20101028|05
00
PV2|||PAIN|||||20101028
GT1|UPDATEDISCHN^OPTWO^||||888 2ND AVE^^LOS
ANGELES^CA^90045|(310)564-6546|||||S |949-84-9494|||||||
IN1|1|MO|||MEDICARE A B|MEDICARE A B|MUTUAL OF OMAHA^BOX 1602-MEDICARE
DEPT^OMAHA^NEBRASKA^92668|429785381|(402)9785381|||||||^^||MO||||||0.00|0||
||101028|dtr|||||||1|1||""||""||||Y|
```

The challenge

```
type Admission struct {
    Type                EventType
    Patient             Patient
    PatientVisit        Visit
    Guarantor           Guarantor
    PatientInsurance    Insurance
}
```

The challenge



Type Model

ACK - General acknowledgment

ADR_A19 - Patient Query - Response

ADT_A01 - Admit/Visit Notification

ADT_A02 - Transfer a Patient

ADT_A03 - Discharge/End Visit

ADT_A04 - Register a Patient

ADT_A05 - Pre-Admit a Patient

ADT_A06 - Change an Outpatient To an Inpatient

ADT_A07 - Change an Inpatient to an Outpatient

ADT_A08 - Update Patient Information

ADT_A09 - Patient Departing - Tracking

ADT_A10 - Patient Arriving - Tracking

ADT_A11 - Cancel Admit/Visit Notification

Type Model

BAR_P02 - Purge Patient Accounts

BAR_P05 - Update Account

BAR_P06 - End account

BAR_P10 - Transmit Ambulatory Payment Classification (APC) Groups

BAR_P12 - Update Diagnosis/Procedure

BPS_O29 - Blood Product Dispense Status Message

ORU_R01 - Unsolicited transmission of an observation message

ORU_R30 - Unsolicited Point-Of-Care Observation Message Without Existing Order – Place An Order

ORU_R31 - Unsolicited New Point-Of-Care Observation Message - Search For An Order

ORU_R32 - Unsolicited Pre-Ordered Point-Of-Care Observation

Message ORU_R01

SEGMENT	OPTIONALITY	REPEATABILITY
MSH - Message Header	R	-
SFT - Software Segment	O	∞
UAC - User Authentication Credential Segment	O	-
> PATIENT_RESULT	R	∞
DSC - Continuation Pointer	O	-

Message ORU_R01

PATIENT RESULT	R	∞
PATIENT	O	-
PID - Patient identification	R	-
PD1 - Patient Additional Demographic	O	-
NK1 - Next of kin / associated parties	O	∞
NTE - Notes and Comments	O	∞

Message ORU_R01 - Segments

Message Header:

```
MSH|^~\&|WAP^WAP||||20090127093601.105-0800||ORU^R01|20090127093601106c5|P|2
.4
```

Patient Identification:

```
PID|1|867509|867509||Van Goe^Edgar^A||||||||||||867509
```

Observation Request:

```
OBR|1|||VITALS^Vital Signs^WAP|||20090127093400.000-0800
```

Observation Result:

```
OBX|1|ST|Heart Rate^Heart Rate^WAP||80|^BPM|||||||20090127093400.000-0800
```

Message ORU_R01 - Patient Fields

THE DEVELOPER'S CONFERENCE

Patient Identification:

```
PID|1|867509|867509||Van Goe^Edgar^A||||||||||||867509
```

- PID.1 - Set ID - PID
- PID.2 - Patient ID
- PID.3 - Patient Identifier List
- PID.4 - Alternate Patient ID - PID
- PID.5 - Patient Name
- PID.6 - Mother's Maiden Name
- PID.7 - Date/Time Of Birth
- PID.8 - Administrative Sex
- PID.9 - Patient Alias
- PID.10 - Race
- PID.11 - Patient Address
- PID.12 - County Code
- PID.13 - Phone Number - Home
- PID.14 - Phone Number - Business
- PID.15 - Primary Language
- PID.16 - Marital Status

The challenge - again

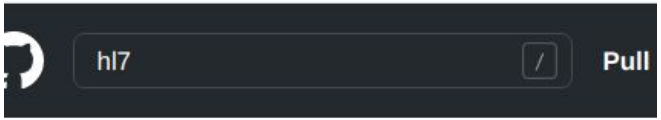
- Parse the message header and delimiters
- Hundreds of different messages
- There’s no real standard coding (implementations)
- Combination of messages
- Combination of request / response

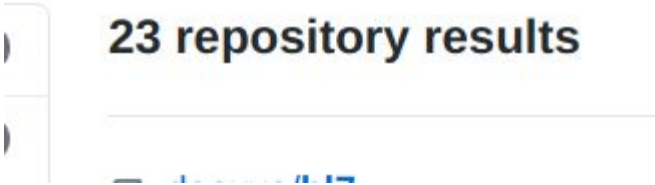
The encoding/json principle

User knows the expected format

```
hl7.Unmarshal(message, &Observation)
```

Fastest solution - Github





Starting point


```
func Reader(message)   []Segments

func (s Segment) Fields() []Fields
```

The challenge - again

- ~~Parse the message header and delimiters~~
- Hundreds of different messages
- There’s no real standard coding (implementations)
- Combination of messages
- Fields using custom space separators
- Data normalization

Data decoder

```go
type stringDecoder func(hl7reader.Fields) string

func textFieldStringDecoder(fields hl7reader.Fields) string {

    var b strings.Builder

    for _, field := range fields {
        for _, comp := range field {
            for _, sub := range comp {
                b.WriteString(sub.String())
                b.WriteString(" ")
            }
        }
    }
    return strings.TrimSpace(b.String())
}
```

```
|Heart Rate^Heart Rate^WAP|

[[Heart Rate]][Heart][Rate][WAP]]

Output: Heart Rate Heart Rate WAP
```

```go
func humanNameStringDecoder(fields hl7reader.Fields) string {

    fNComp, ok := fields.GetSubComponent(0, 0, 0)
    familyName := ""
    if ok {
        familyName = fNComp.String()
    }

    var b strings.Builder

    for _, field := range fields {
        for i, comp := range field {
            //Skip family name
            if i == 0 {
                continue
            }
            for _, sub := range comp {
                b.WriteString(sub.String())
                b.WriteString(" ")
            }
        }
    }
    b.WriteString(familyName)
    return b.String()
}
```

```
|Van Goe^Edgar^A|

[[[Van][Goe]][Edgar][A]]

Output: Edgar A Van Goe
```

Data decoder

```
func selectDecoderFunction(td typeData) stringDecoder {
    switch td.valueType {
    case "xpn":
        return humanNameStringDecoder
    case "cx":
        return textFieldStringDecoder
    default:
        return textFieldStringDecoder
    }
}
```

Data decoder

```
func decodeToString(fields hl7reader.Fields, td typeData) string {

    decoderFunc := selectDecoderFunction(td)
    return decoderFunc(fields)
}
```

The challenge - again

- ~~Parse the message header and delimiters~~
- Hundreds of different messages
- There’s no real standard coding (implementations)
- Combination of messages
- ~~Fields using custom space separators~~
- ~~Data normalization~~

Client defined data decoding structure

```
type Patient struct {
    PatientID   string `hl7:"2,cx"`
    AlternateID string `hl7:"3,cx"`
    Name        string `hl7:"5,xpn"`
}

type Observation struct {
    Type        string `hl7:"3,cx"`
    Measurament string `hl7:"4,cx"`
    Value       string `hl7:"5,cx"`
    Unit        string `hl7:"6,cx"`
}
```

Reflection to parse tag

```go
func UnmarshalSegment(segment hl7reader.Segment, dst interface{}) error {

    v := reflect.Indirect(reflect.ValueOf(dst))

    for i := 0; i < v.NumField(); i++ {

        field := v.Type().Field(i)
        tag := field.Tag.Get(tagName)

        if tag != "" {
            err := decodeSegmentValue(segment, v, field.Name, tag)
            if err != nil {
                return err
            }
        }
    }

    return nil
}
```

Reflection to parse tag

```
func decodeSegmentValue(segment hl7reader.Segment,
    sctValue reflect.Value,
    fieldName string,
    tagValue string,
) error {

    f := sctValue.FieldByName(fieldName)

    dataType := parseTag(tagValue)

    actualValue, err := extractDataFromSegment(segment, dataType)
    if err != nil {
        return err
    }
    f.SetString(actualValue)
    return nil
}
```

```
"5,cx" = dataType {
    position: 5,
    type: cx
}
```

```go
func readMessage(message string) {

    msg, err := hl7.Read(message)

    if err == nil {
        fmt.Println(msg)
        fmt.Println(msg.Type())
        oruMsg := msg.(*hl7.ORU_R01)
        fmt.Println(oruMsg.Patient)
    }
}
```

```
&{0xc00007eed0 [{Heart Rate Heart Rate WAP  80 BPM} {Temperature Temperature WAP 1 98.6 °F} {Temperature Temperature WAP 2 97.5 °F} {SPO2 SPO2 WAP  97 %} {CO2 (In) CO2 (In) WAP  0.0 %} {CO2 (Ex) CO2 (Ex) WAP  5.0 %} {RR/BR RR/BR WAP 12 Br/M} {PVC PVC WAP  0.0 PVC/Min}]}

ORU_R01

&{867509 867509 Edgar A Van Goe}
```

The challenge - again

- ~~Parse the message header and delimiters~~
- ~~Hundreds of different messages~~
- There’s no real standard coding (implementations)
- ~~Combination of messages~~
- ~~Fields using custom space separators~~
- ~~Data normalization~~

No real standard


```
|Van Goe^Edgar^A|

[[[Van][Goe]][Edgar][A]]

Output: Edgar A Van Goe
```

```
|Van^Goe^Edgar^A|

[[Van][Goe][Edgar][A]]

Output: Edgar A Van Goe
```

```
|Van Goe Edgar A|

[[Van][Goe][Edgar][A]]

Output: Edgar A Van Goe
```

```
|Van Goe Edgar^A^MR|

[[[Van][Goe][Edgar]][A]]

Output: Edgar A Van Goe
```

Client defined decoders

```
func defaultDecoders() map[string]stringDecoder {
    decs := map[string]stringDecoder{}
    decs["xpn"] = humanNameStringDecoder
    decs["cx"] = textFieldStringDecoder
    return decs
}

func (d *decoder) selectDecoderFunction(valueType string) stringDecoder {

    if dec, ok := d.stringDecoders[valueType]; ok {
        return dec
    } else {
        return fallbackDecoder
    }
}
```

Client defined decoders

```
func WithDecoders(decoders map[string]StringDecoder) *decoder {
    dd := defaultDecoders()

    for k, v := range decoders {
        dd[k] = v
    }
    return &decoder{
        dd,
    }
}
```

Client defined decoders

```
func prefixedNameDecoder(prefix string) hl7.StringDecoder {
    return func(fields hl7reader.Fields) string {
        txt := hl7.HumanNameStringDecoder(fields)
        return prefix + " " + txt
    }
}
```

Client defined decoders

```
func readDecoderMessage(message string) {

    customDec := map[string]hl7.StringDecoder{}
    customDec["xpn"] = prefixedNameDecoder("Patient Name is: ")
    msg, err := hl7.ReadWithDecoders(customDec, message)

    if err == nil {
        fmt.Println(msg)
        fmt.Println(msg.Type())
        oruMsg := msg.(*hl7.ORU_R01)
        fmt.Println(oruMsg.Patient)
    }
}
```

```
&{867509 867509 Patient Name is:  Edgar A Van Goe}
```

The challenge - again

- Parse the message header and delimiters
- Hundreds of different messages
- There’s no real standard coding (implementations)
- Combination of messages
- Fields using custom space separators
- Data normalization

The challenge - again

- Raw message parsing
- Message content variation
- Payload combination, union, merge
- Custom encoding
- Data normalization

Referências

exemplos: https://github.com/zamariola

Referências

- https://project-awesome.org/kakoni/awesome-healthcare
- https://medium.com/@okulewicz.amelia/chaos-in-the-standardisation-how-to-handle-hl7-96fe66c657c5
- https://hapifhir.github.io/hapi-hl7v2/
- https://www.welchallyn.com/content/dam/welchallyn/documents/sap-documents/LIT/80018/80018556LITPDF.pdf
- https://www.fknsrs.biz/blog/golang-hl7-library.html

Thank you!

www.zamariola.com.br

Support Slides

SEGMENT	OPTIONALITY	REPEATABILITY
MSH - Message Header	R	-
PATIENT RESULT	R	∞
PATIENT	O	-
PID - Patient identification	R	-
PD1 - Patient Additional Demographic	O	-
NK1 - Next of kin / associated parties	O	∞
NTE - Notes and Comments	O	∞
VISIT	O	-
ORDER OBSERVATION	R	∞
ORC - Common Order	O	-
OBR - Observation Request	R	-
NTE - Notes and Comments	O	∞
CTD - Contact Data	O	-
OBSERVATION	R	∞
OBX - Observation/Result	O	-
NTE - Notes and Comments	O	∞

FIELD	LENGTH	DATA TYPE	OPTIONALITY	REPEATABILITY	TABLE
PID.1 - Set ID - PID	4	SI	O	-	
PID.2 - Patient ID	20	CX	B	-	
PID.3 - Patient Identifier List	250	CX	R	∞	
PID.4 - Alternate Patient ID - PID	20	CX	B	∞	
PID.5 - Patient Name	250	XPN	R	∞	
PID.6 - Mother's Maiden Name	250	XPN	O	∞	
PID.7 - Date/Time Of Birth	26	TS	O	-	
PID.8 - Administrative Sex	1	IS	O	-	0001
PID.9 - Patient Alias	250	XPN	B	∞	
PID.10 - Race	250	CE	O	∞	0005
PID.11 - Patient Address	250	XAD	O	∞	
PID.12 - County Code	4	IS	B	-	0289
PID.13 - Phone Number - Home	250	XTN	O	∞	
PID.14 - Phone Number - Business	250	XTN	O	∞	
PID.15 - Primary Language	250	CE	O	-	0296
PID.16 - Marital Status	250	CE	O	-	0002

FIELD	LENGTH	DATA TYPE	OPTIONALITY	REPEATABILITY	TABLE
OBX.1 - Set ID - OBX	4	SI	O	-	
OBX.2 - Value Type	2	ID	C	-	0125
OBX.3 - Observation Identifier	250	CE	R	-	
OBX.4 - Observation Sub-Id	20	ST	C	-	
OBX.5 - Observation Value	65536	VARIES	C	∞	
OBX.6 - Units	250	CE	O	-	
OBX.7 - References Range	60	ST	O	-	
OBX.8 - Abnormal Flags	5	IS	O	5	0078
OBX.9 - Probability	5	NM	O	-	
OBX.10 - Nature of Abnormal Test	2	ID	O	∞	0080
OBX.11 - Observation Result Status	1	ID	R	-	0085
OBX.12 - Date Last Observation Normal Value	26	TS	O	-	
OBX.13 - User Defined Access Checks	20	ST	O	-	
OBX.14 - Date/Time of the Observation	26	TS	O	-	
OBX.15 - Producer's ID	250	CE	O	-	
OBX.16 - Responsible Observer	250	XCN	O	∞	