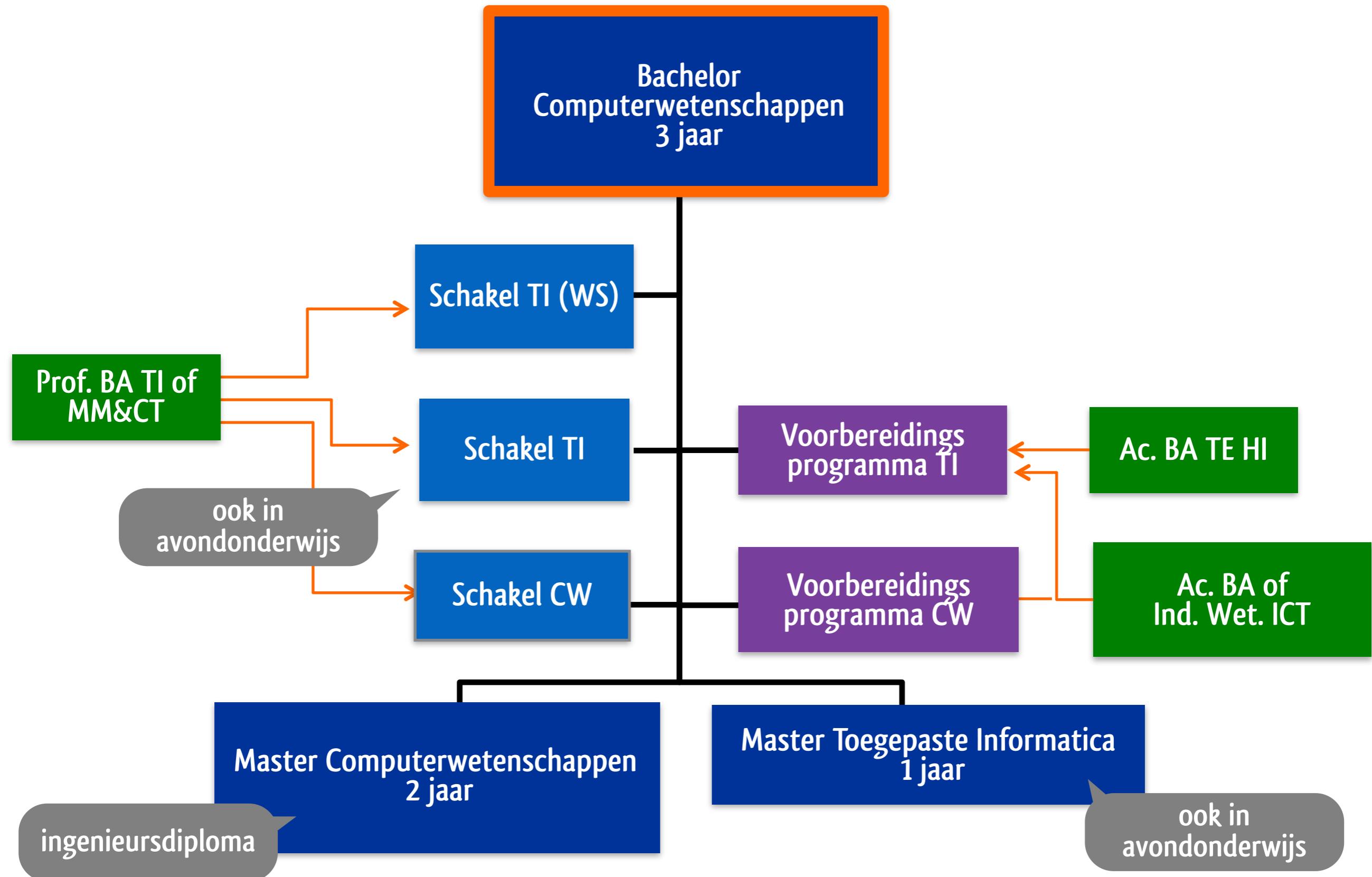
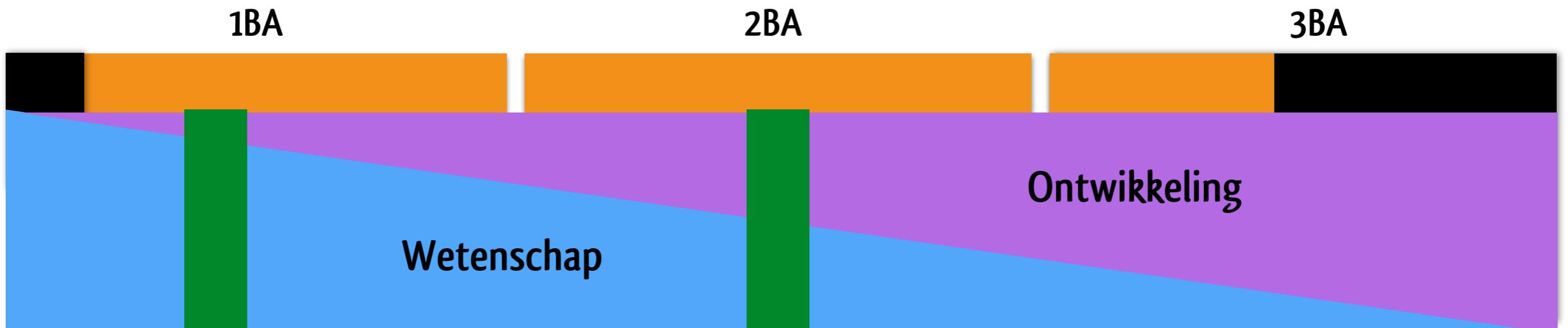


Bachelor Computerwetenschappen



Wetenschap + programmeertalen + ontwikkeling



Structuur van Computerprogramma's 1

Algoritmen en Datastructuren 1

Logica en Formele Systemen

Wiskunde: Calculus en Lineaire Algebra

Interpretatie van Computerprogramma's 1

Besturingssystemen en Systeemfundam.

Databanken

Programmeerproject 1

Basisvaardigheden - User Interfaces -
Privacy op Internet

Discrete Wiskunde

Algoritmen en Datastructuren 2

Automaten en Berekenbaarheid

Kansrekening en Statistiek

Artificiële Intelligentie

Computersystemen

Structuur van Computerprogramma's 2

Object-gericht Programmeren

Object-gericht Modelleren

Communicatievaardigheden

Programmeerproject 2

Wetenschappelijk Rekenen
Parallelisme en Distributie

Software Engineering

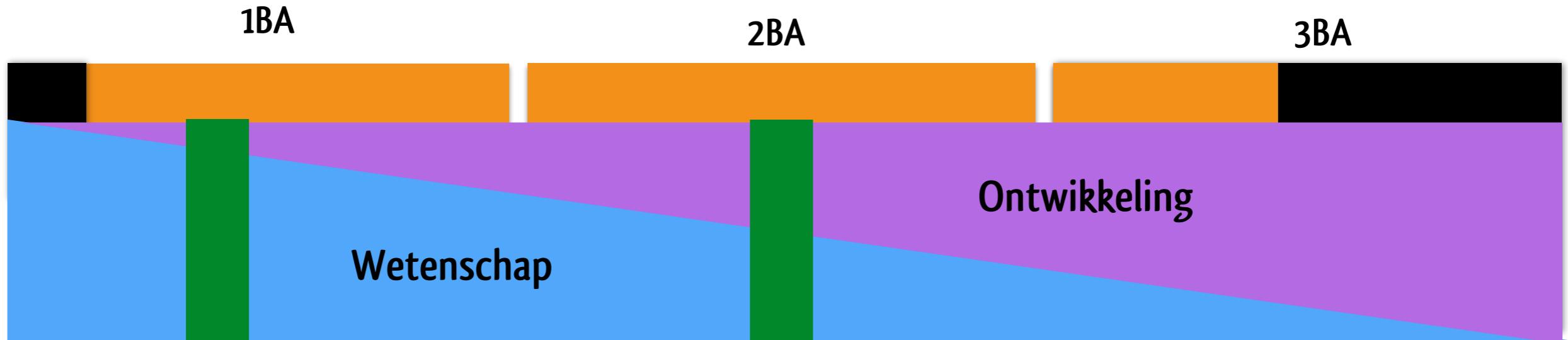
Economie en Bedrijfsleven

Bachelorproef

Keuze (33 ECTS)

- BA bereidt voor op MATI of MA CW ingenieursstudie
- Grote ruimte voor keuze en profiling in 3BA (33ECTS)

Wetenschap + programmeertalen + ontwikkeling



Structuur van Computerprogramma's 1

Algoritmen en Datastructuren 1

Logica en Formele Systemen

Wiskunde: Calculus en Lineaire Algebra

Interpretatie van Computerprogramma's 1

Besturingssystemen en Systeemfundam.

Databanken

Programmeerproject 1

Basisvaardigheden - User Interfaces -

Privacy op Internet

Discrete Wiskunde

Algoritmen en Datastructuren 2

Automaten en Berekenbaarheid

Kansrekening en Statistiek

Artificiële Intelligentie

Computersystemen

Structuur van Computerprogramma's 2

Object-gericht Programmeren

Object-gericht Modelleren

Communicatievaardigheden

Programmeerproject 2

Wetenschappelijk Rekenen

Parallelisme en Distributie

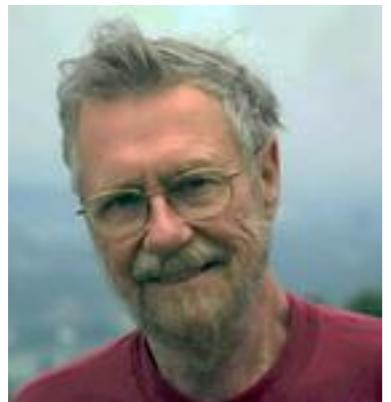
Software Engineering

Economie en Bedrijfsleven

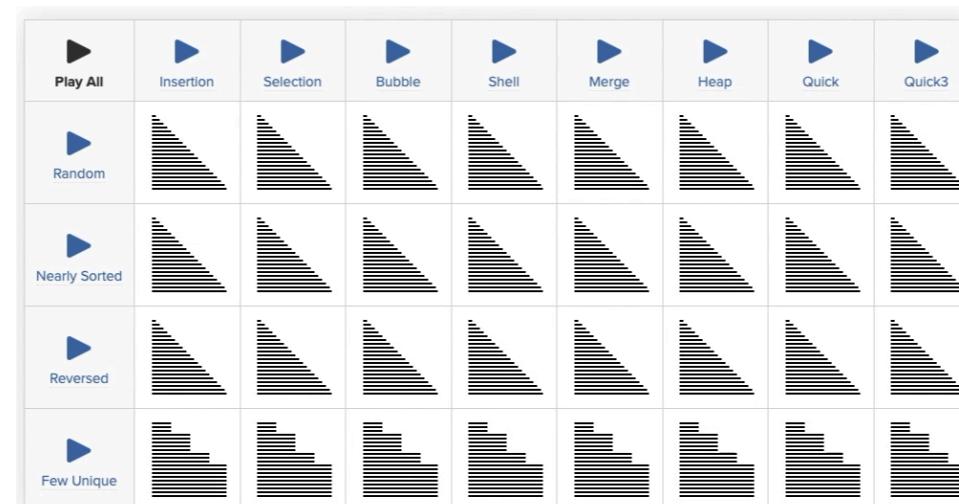
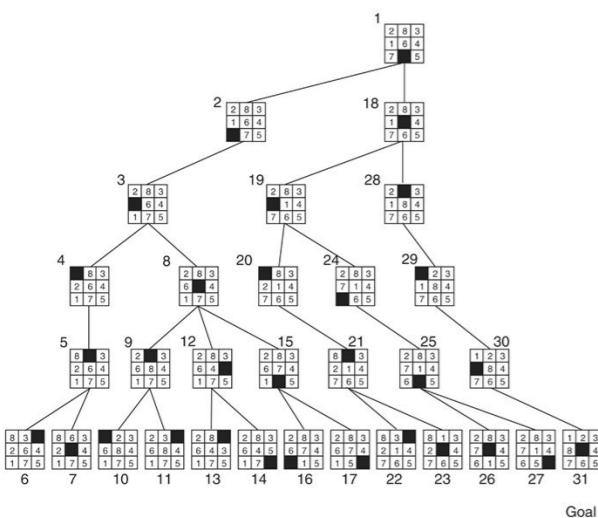
Bachelorproef

Keuze (33 ECTS)

Computerwetenschappen



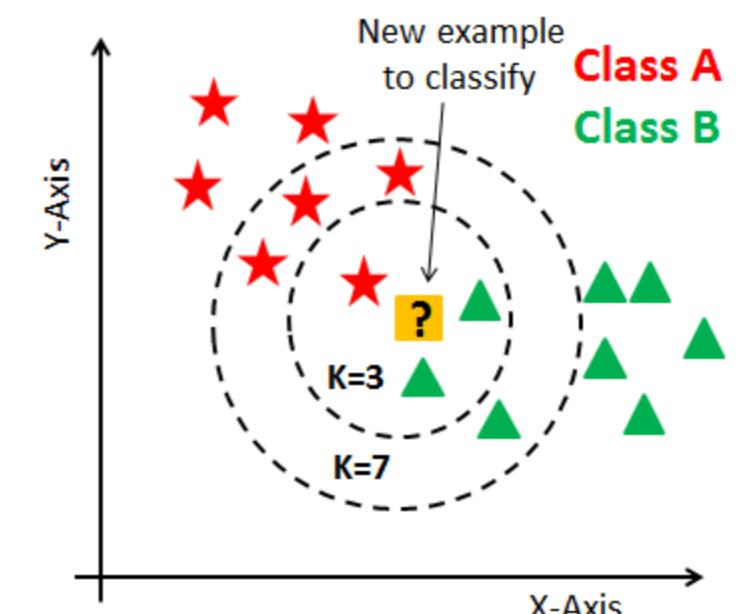
“Computer science is no more about computers than astronomy is about telescopes”
(Edsger W. Dijkstra, 1930-2002)



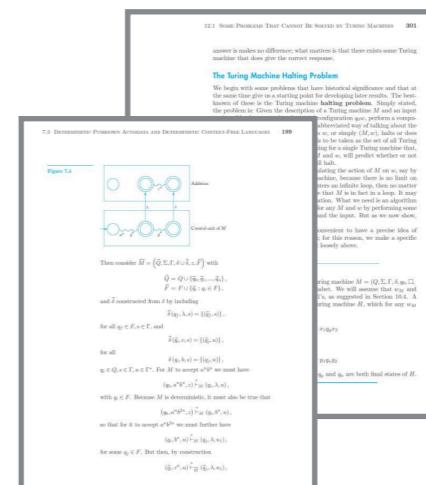
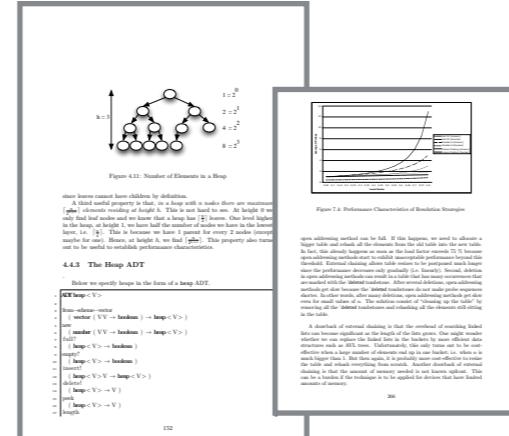
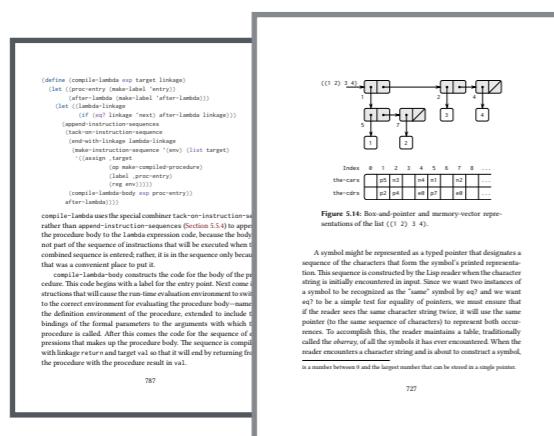
zoeken

sorteren

classificeren



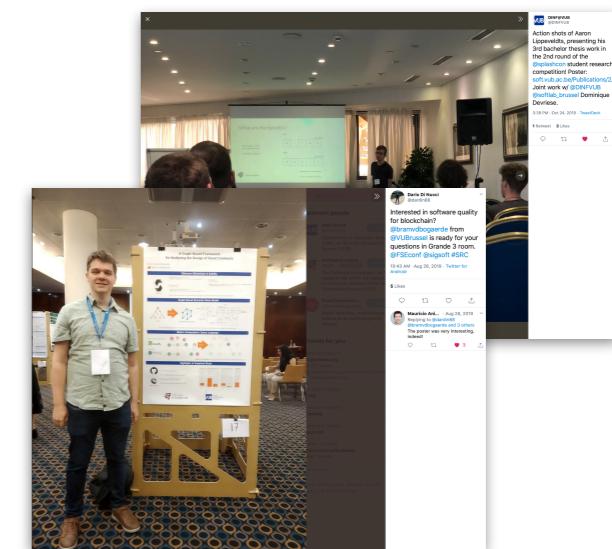
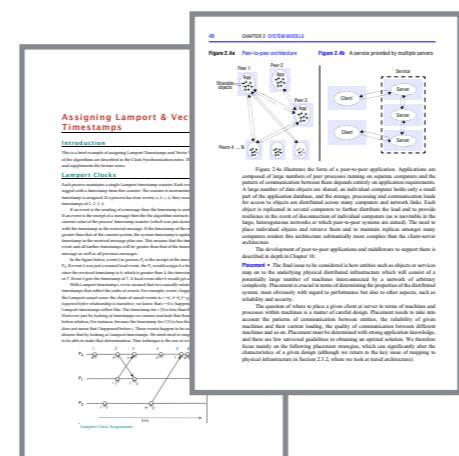
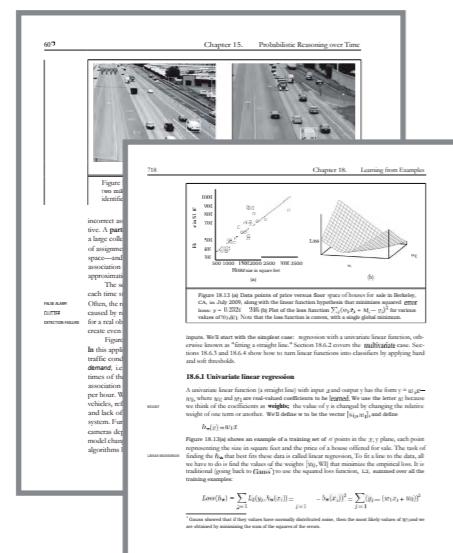
Computerwetenschappen aan de VUB



interpretatie van computerprogramma's

algoritmen & datastructuren

automaten & berekenbaarheid



artificiële intelligentie

parallelisme & distributie

bachelor thesis

- samen met wiskunde-vakken, een solide wetenschappelijke basis
- proeven van onderzoek in bachelor thesis

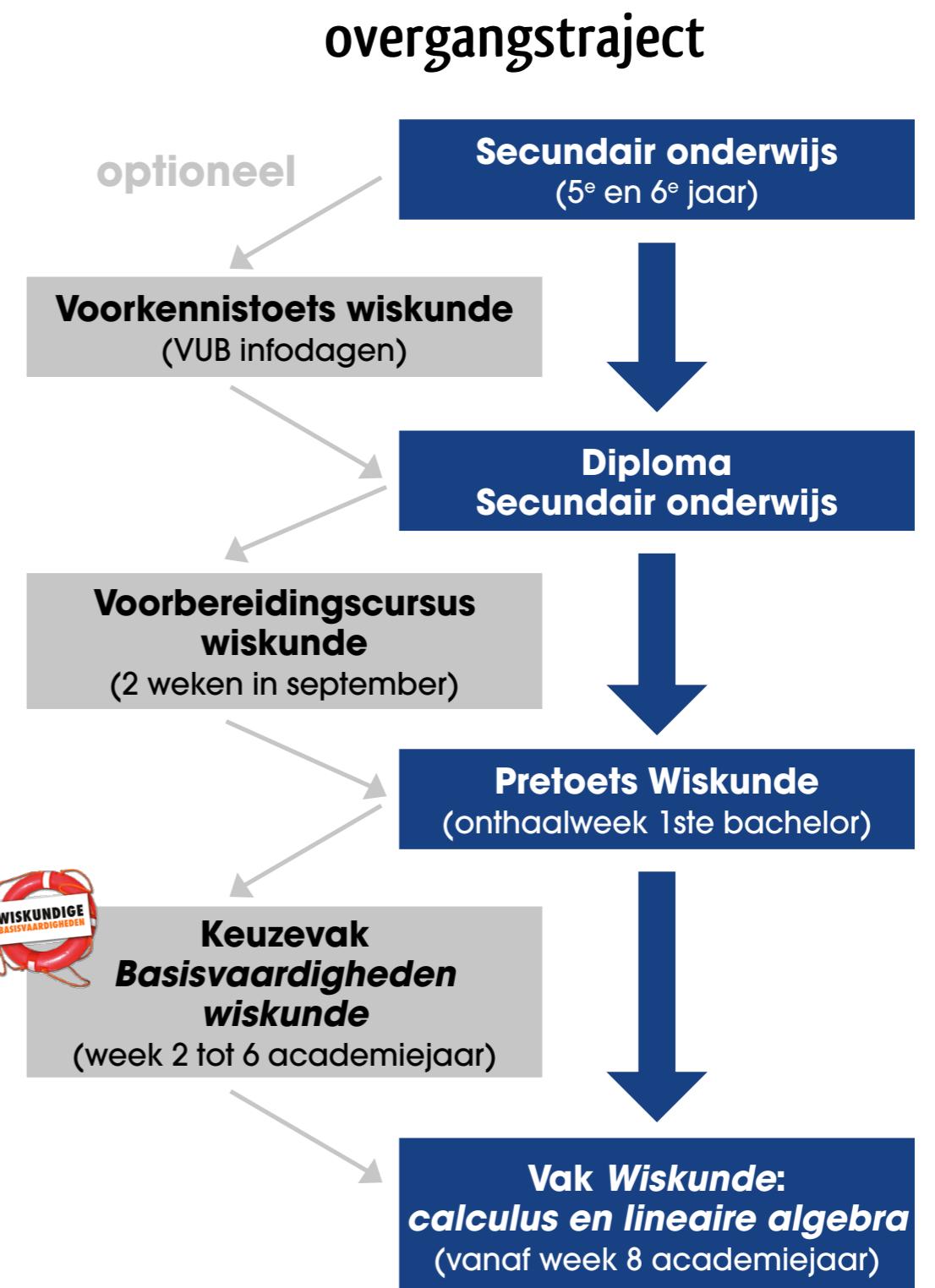
Wiskunde ter ondersteuning van computerwetenschap

1ste BA
Logica en Formele Systemen
Wiskunde: Calculus en Lineaire Algebra

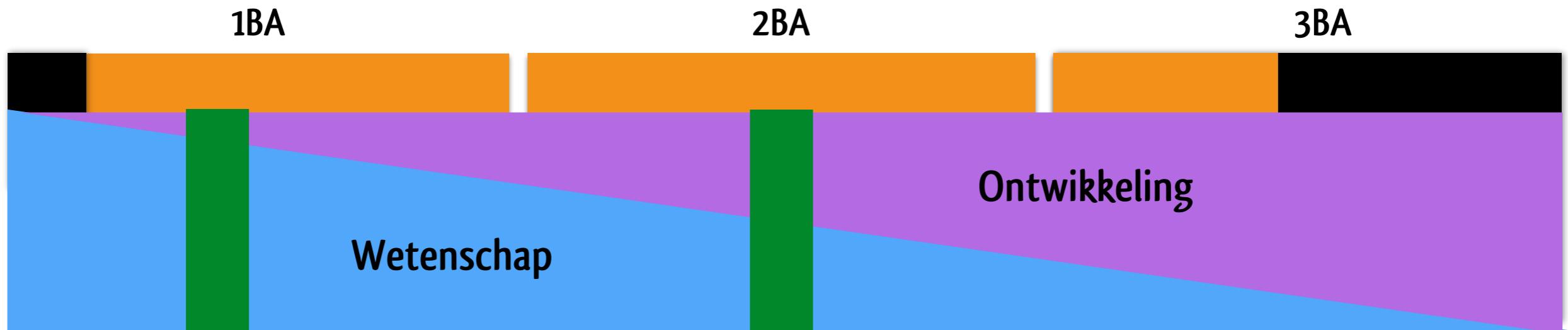
2de BA
Discrete Wiskunde
Kansrekening en Statistiek

3de BA
Wetenschappelijk Rekenen

<https://www.vub.be/voorbereiden>



Wetenschap + programmeertalen + ontwikkeling



Structuur van Computerprogramma's 1

Algoritmen en Datastructuren 1

Logica en Formele Systemen

Wiskunde: Calculus en Lineaire Algebra

Interpretatie van Computerprogramma's 1

Besturingssystemen en Systeemfundam.

Databanken

Programmeerproject 1

Basisvaardigheden - User Interfaces -

Privacy op Internet

Discrete Wiskunde

Algoritmen en Datastructuren 2

Automaten en Berekenbaarheid

Kansrekening en Statistiek

Artificiële Intelligentie

Computersystemen

Structuur van Computerprogramma's 2

Object-gericht Programmeren

Object-gericht Modelleren

Communicatievaardigheden

Programmeerproject 2

Wetenschappelijk Rekenen

Parallelisme en Distributie

Software Engineering

Economie en Bedrijfsleven

Bachelorproef

Keuze (33 ECTS)

Programmeertalen

```
(define (fac n)
  (if (= n 0)
      1
      (* n (fac (- n 1)))))

(fac 5)
```

Scheme

```
fact :: Integer -> Int
fact 0 = 1
fact n = n * fact (n - 1)
fact 5
```

Haskell

```
public class Factorial {
    public static long factorial( int n ) {
        if( n <= 1 )
            return 1;
        else
            return n * factorial( n - 1 );
    }
    public static void main( String [ ] args {
        System.out.println( factorial( 5 ) );
    }
}
```

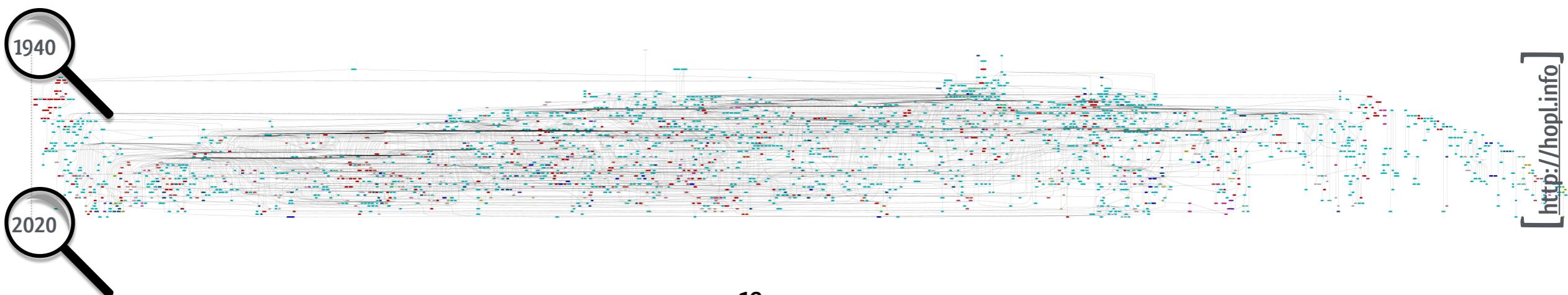
Java

```
def fact(x):
    return (1 if x==0 else x * fact(x-1))

print(fact(5))
```

Python

volgens bepaalde schattingen >5000 in gebruik



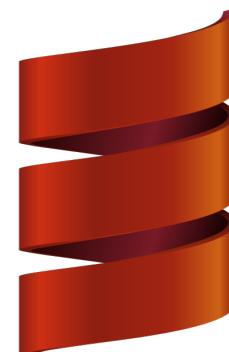
Programmeertalen aan de VUB



Scheme

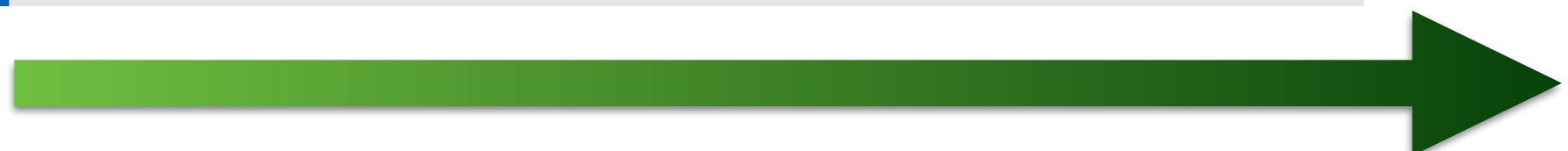


C



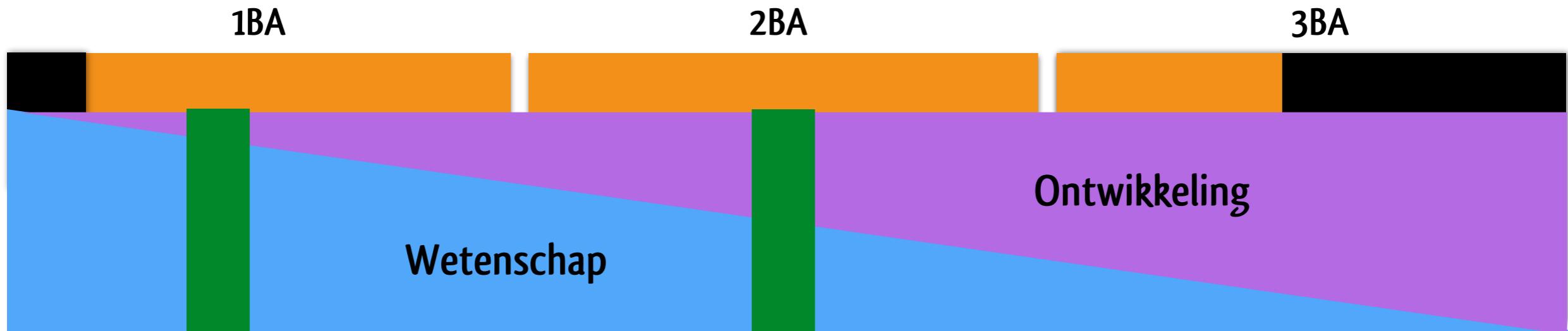
Scala

paradigma	multi-paradigmatisch	procedureel	object-gericht
typering	dynamisch	zwak statisch	sterk statisch
geheugenbeheer	automatisch	manueel	automatisch
jaar	1ste BA	2de BA	vanaf 2de BA



- doordachte keuze die het volledige **spectrum aan paradigma's** afdekt
 - geen voorkennis verondersteld, zelfde talen gebruikt in andere vakken
 - loopbaan van 44 jaar: zelfstandig nieuwe talen kunnen leren
- belangrijke onderzoeksgroep in programmeertalen
 - aandacht voor ontwerp en implementatie van programmeertalen

Wetenschap + programmeertalen + ontwikkeling

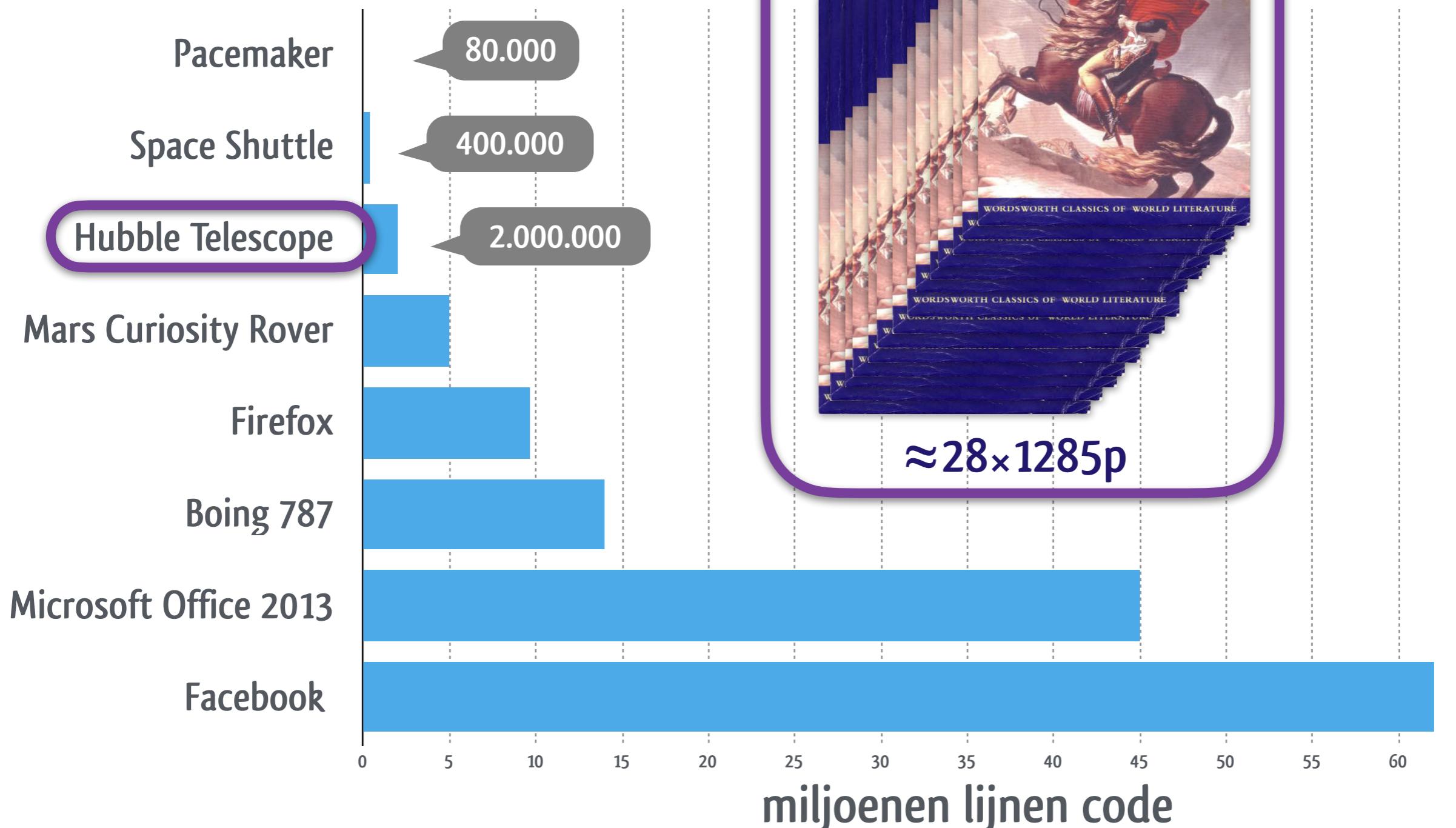


Structuur van Computerprogramma's 1
Algoritmen en Datastructuren 1
Logica en Formele Systemen
Wiskunde: Calculus en Lineaire Algebra
Interpretatie van Computerprogramma's 1
Besturingssystemen en Systeemfundam.
Databanken
Programmeerproject 1
Basisvaardigheden - User Interfaces -
Privacy op Internet

Discrete Wiskunde
Algoritmen en Datastructuren 2
Automaten en Berekenbaarheid
Kansrekening en Statistiek
Artificiële Intelligentie
Computersystemen
Structuur van Computerprogramma's 2
Object-gericht Programmeren
Object-gericht Modelleren
Communicatievaardigheden
Programmeerproject 2

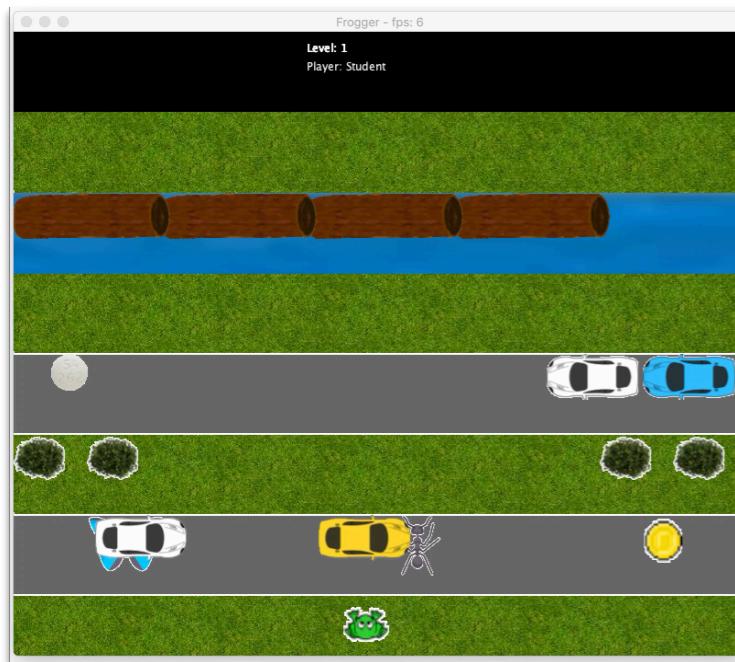
Wetenschappelijk Rekenen
Parallelisme en Distributie
Software Engineering
Economie en Bedrijfsleven
Bachelorproef
Keuze (33 ECTS)

Software-ontwikkeling

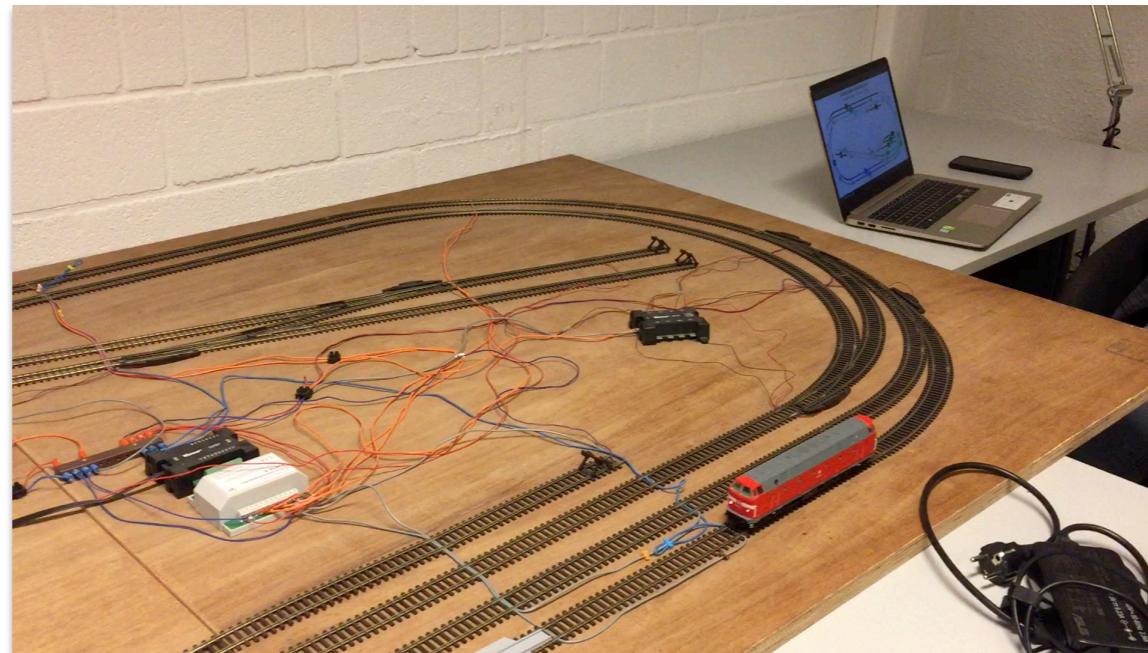


[https://docs.google.com/spreadsheets/d/1s9u0uprmvwR2fkRqxl4W5Wfomimmk9pwGtK4Dn_Ul/edit#gid=5]

Software-ontwikkeling aan de VUB



1ste BA



2de BA



3de BA

retrospelletje

~2000 LOC

individueel

van nul

programmeren

simulatie Infrabel+NMBS

~3000 LOC

individueel

van nul

algoritmes + wereld eromheen

editor interieurplannen

> 10.000 LOC

groepsverband

uitbreiding bestaande code

professionalisatie

33ECTS keuzevakken in 3de bachelor

verdiepend

- 6ECTS credits - Interpretation of Computer Programs 2
- 3ECTS credits - Digitale wiskunde
- 6ECTS credits - Machine Learning
- 6ECTS credits - Multimedia Processing Tools
- 6ECTS credits - IT Networks
- 6ECTS credits - Web Technologies
- 3ECTS credits - Bachelor onderzoeksstage
- 3ECTS credits - Privacy, veiligheid en eigendom op internet
- 3ECTS credits - Gebruikersinterfaces
- 3ECTS credits - Evolution of Software Languages

onderwijs

- 6ECTS credits - Leren van individuele leerlingen
- 6ECTS credits - Urban Education
- 3ECTS credits - Onderwijsociologie en onderwijsbeleid

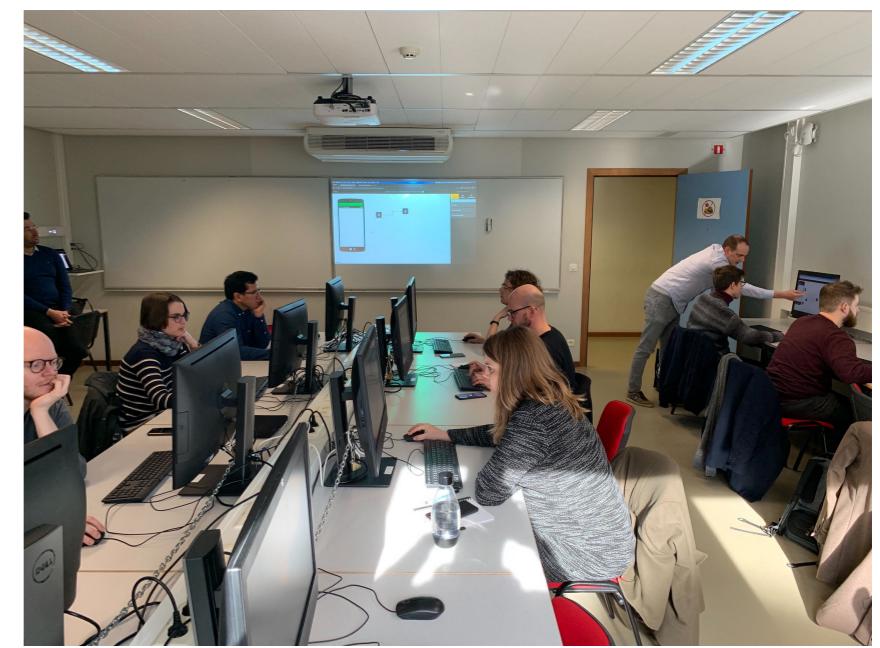
verbredend

- 3ECTS credits - Fysica: inleiding mechanica
- 3ECTS credits - Evolutie
- 3ECTS credits - Biotechnologie
- 3ECTS credits - Business Aspects of Software Industry
- 3ECTS credits - Academic English 2
- 3ECTS credits - Academic English I
- 3ECTS credits - Redelijk eigenzinnig. Nadenken over Mens en Maatschappij
- 6ECTS credits - Inleiding tot de algemene taalwetenschap
- 3ECTS credits - De grote stromingen in de wijsbegeerde van de oudheid tot het heden
- 6ECTS credits - Cognitieve psychologie I
- 6ECTS credits - Marketingcommunicatie
- 6ECTS credits - Chemie: bouw van de materie en chemische reacties I
- 6ECTS credits - Inleiding tot het recht
- 6ECTS credits - Management
- 6ECTS credits - Wiskunde voor Data Science
- 6ECTS credits - Marketing
- 6ECTS credits - Algemene biologie
- 6ECTS credits - Sociologie I
- 3ECTS credits - Sociale psychologie I
- 6ECTS credits - Digitaal mediagebruik
- 3ECTS credits - Inleiding tot het bedrijfsbeheer
- 6ECTS credits - Geo-informatiekunde
- 4ECTS credits - Marketing

Organisatie en ondersteuning

- **gecoördineerde** uurroosters en examenroosters
- **regelmatige** **taken** in 1BA:
 - elke week één taak (1 per 3SP)
 - centrale coördinatie
- **tussentijdse evaluaties** in week 7:
 - logica en formele systemen
 - structuur van computerprogramma's 1
 - (basisvaardigheden wiskunde)
- ondersteuning vanuit **SBC**:
 - vakspecifieke studiebegeleidster
 - VIP-classes
 - opvolggesprekken
 - massaprogrammeren

Infrastructuur: hoorcolleges en werkcolleges



Infrastructuur: afstandsonderwijs

Two screenshots of a Panopto video library interface are shown. The left screenshot shows a list of recorded lectures in a folder named "1920 - Software Architectures - 005419". The right screenshot shows a specific lecture titled "1.1 Meta-circulaire interpretatie (deel 1)".

Left Screenshot (List of recordings):

- 1. Tour of Scala: Function... Oriented Basics (1:45:32)
- 1.4.a Family Polymorphism Problem (1:38:26)
- 1.4.b Expression Problem & 2. Docu... Architectures (1:38:29)
- 3.2 Component-and-Connector Patterns (1:14:57)
- 3.2 C&C patterns: Shared Data, Cl... Pipe-and-Filter (45:33)
- 3.2 C&C patterns: Pipe-a... Akka Streams (part 1) (51:46)
- 3.2 C&C patterns: Pipe-... Model-View-Controller (4 months ago)
- 3.2 C&C patterns: Micro-services ill... Akka (part 2) (58:50)
- 3.2 C&C patterns: Publish-Subscribe, ... Tier (46:11)
- 4. Actor-based Design Patterns for Micro-Service Architectures (49:24)
- 4.3 Flow Control Patterns (51:51)
- 4.3 Flow Control Patterns: Pull Pattern, Managed (3 months ago)
- 4.4 Fault-Tolerance Patterns... Circuit Breaker Telenet (2 months ago)

Right Screenshot (Specific Lecture):

Lecture Title: 1.1 Meta-circulaire interpretatie (deel 1)

Content Summary:

```
(define (eval exp env)
  (cond ((self-evaluating? exp) exp)
        ((variable? exp) (lookup-variable-value exp env))
        ((quoted? exp) (text-of-quotation exp))
        ((assignment? exp) (eval-assignment exp env))
        ((definition? exp) (eval-definition exp env))
        ((if? exp) (eval-if exp env))
        ((lambda? exp)
         (make-procedure (lambda-parameters exp)
                        (lambda-body exp)
                        env))
        ((begin? exp)
         (eval-sequence (begin-actions exp) env))
        ((cond? exp) (eval (cond->if exp) env))
        ((application? exp)
         (apply (eval (operator exp) env)
                (list-of-values (operands exp) env))))
        (else
         (error "Unknown expression type -- EVAL" exp))))
```

Annotations on the right:

- evaluatie van een expressie in een omgeving** (Evaluation of an expression in an environment)
- analyseert de syntax van de te evalueren expressie** (Analyzes the syntax of the expression to be evaluated)
- rechtstreekse aanmaak van waarde** (Direct creation of value)
- evaluatie van eenvoudigere expressie** (Evaluation of simpler expression)
- expressiesoorten boven deze tak zijn "speciaal"** (Expression types above this branch are "special")
- delegatie naar gespecialiseerde procedures** (Delegation to specialized procedures)

Player Controls:

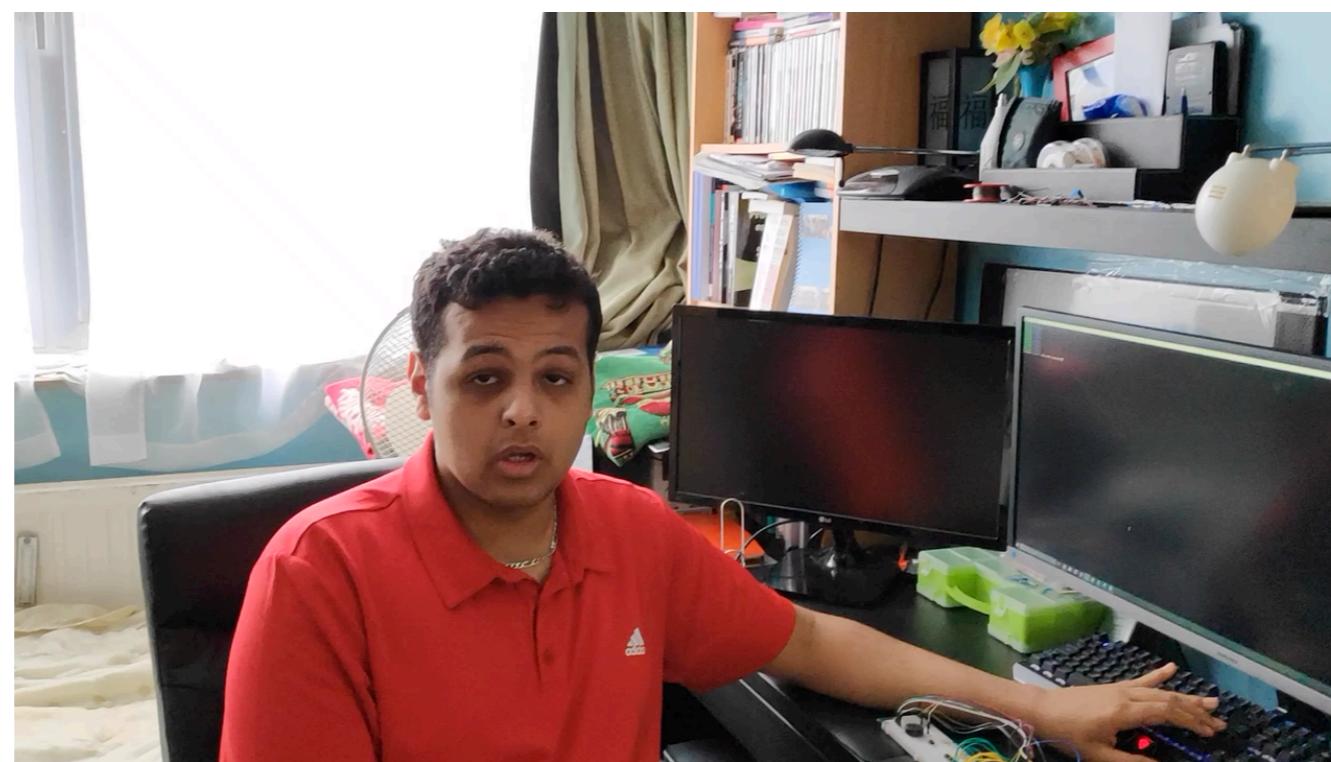
- Speed: 1x
- Quality: 100%
- Volume: 100%
- Time: 1:09:56 / 1:08:18

Infrastructuur: in alle maten en kleuren

2018-2019 Kevin Coorens



2018-2019 Hicham Azmani



2018-2019 Jasper Zucka

Voorbeeld lessenrooster

1^{ste} jaar Computerwetenschappen

Voorbeeld week 5 (oktober)

	Maandag	Dinsdag	Woensdag	Donderdag	Vrijdag
8h-9h					
9h-10h				Basisvaardigheden wiskunde (praktijk) <i>Vb keuzevak</i>	Basisvaardigheden wiskunde (praktijk) <i>Vb keuzevak</i>
10h-11h					
11h-12h	Structuur van computerprogramma's I (praktijk)			Structuur van computerprogramma's I (theorie)	
12h-13h					
13h-14h		Algoritmen en datastructuren I (theorie)			
14h-15h	Structuur van computerprogramma's I (theorie)		Structuur van computerprogramma's I (praktijk)	Basisvaardigheden wiskunde (theorie)	Algoritmen en datastructuren I (theorie)
15h-16h		Logica en formele systemen (theorie)			
16h-17h	Logica en formele systemen (praktijk)		Algoritmen en datastructuren I (praktijk)		
17h-18h					

Focusgesprek met studenten tijdens kwaliteitsbewaking



Alle bevraagde studenten
zouden opnieuw voor de
opleiding kiezen!

- **Interessant programma en goede hoorcolleges**
- Goede volgtijdelijkheid en **samenhang van de vakken**
- **Kleine groepen**
- Docenten en assistenten zijn zeer **toegankelijk en aanspreekbaar**
- Veel **projectwerk** doorheen het hele jaar
- Goede **ondersteuning** door het SBC
- Goede **fundamentele aanpak** van programmeren, makkelijk andere talen te leren
- Het schakelprogramma is goed opgebouwd en is goed verdeeld
- Wat verwacht wordt voor een examen is meestal duidelijk
- Studenten worden fair en objectief beoordeeld

Wist je dat?



pionier sinds 1978

.AGORIA

30.000 ICT vacatures voor ICT
experten in 2018



Felix Van de Maele

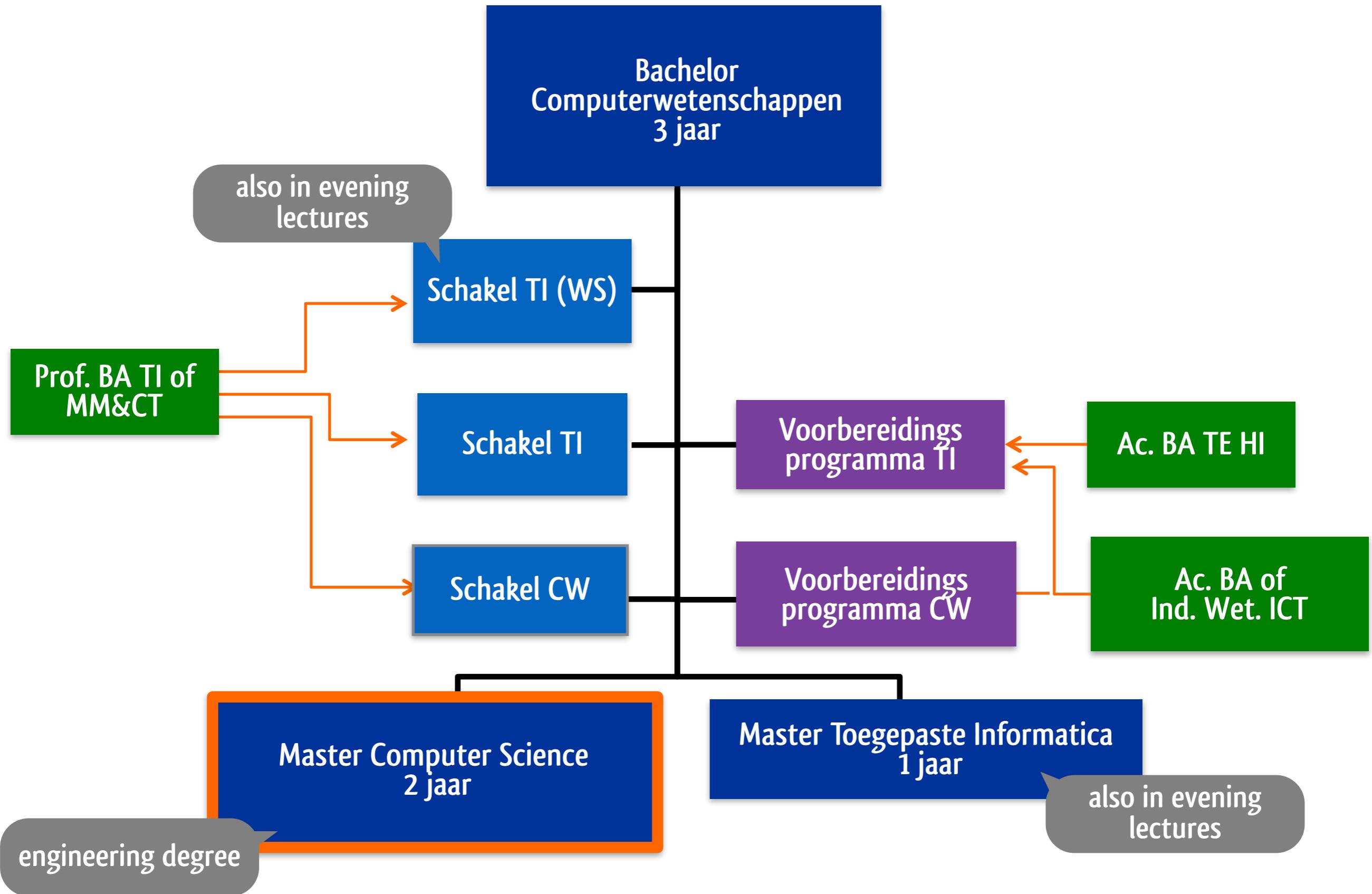


Pattie Maes



bruto gemiddeld van 4.666€ voor een ontwikkelaar
tot 7.160€ voor een manager in 2016

Master of Applied Sciences and Engineering in Computer Science



Architecture of the programme

30 ECTS Mandatory Common

Methods of Scientific Research
Scientific Integrity
Information Theory
Theory of Computation
Open Information Systems
Declarative Programming
Software Architectures

30 ECTS Specialisation

Mandatory + Electives

30 ECTS Electives Common



Mobility
Window

6 ECTS

Research Training

24 ECTS

Master Thesis in
Specialisation

- strongly embedded in ongoing research activities of the labs
- prepares for role in CS research and development, in academia and industry

Example course from the core: Software Architectures

FTGO μ-services architecture

The diagram illustrates a monolithic system that has been horizontally partitioned into several microservices. At the top, a mobile application (Consumer) and a Courier interact with an API Gateway. The API Gateway routes requests to various services: Order Service, Restaurant Service, Kitchen Service, and Delivery Service. Each service is represented by a hexagon containing a REST API icon and a database icon. The Restaurant Service also interacts with a Restaurant Web UI. Annotations explain that the API Gateway routes requests from mobile applications to services, and that services have APIs.

- monolith distributed vertically into services that ...
- each service provides and consumes functionality

FTGO μ-services: messaging (2/2)

A photograph of a deck of cards titled "Enterprise Integration Patterns". The cards are designed to look like playing cards and feature integration patterns such as "Publish-Subscribe Channel", "Splitter", "Aggregator", and "Scatter-Gather". The background shows a book titled "Enterprise Integration Patterns" and a small screen displaying a news article about the 20th anniversary of patterns.

[Zimmermann et al., IEEE Software 2016]

But resilience remains difficult to get right ...

```
4 trait Event
5 case class Plus(amount: Int)
6 case class PlusEvent(amount : Int) extends Event
7 case class CountCommand(id : Long, amount : Int)
8 case class ConfirmEvent(id : Long) extends Event
9 case class Confirm(id : Long)
10
11 class GuaranteedDeliveryActor(ref: ActorRef)
12 extends PersistentActor with AtLeastOnceDelivery {
```

A diagram showing five resilience strategies for message handling, each represented by a colored bar: "regular message handler" (grey), "recovery handler used to rehydrate state from journal" (orange), "message send with at-least-once delivery" (yellow), "stop resending" (blue), and "message processing acknowledgement sent back" (green).

Ongoing research

21

- large-scale systems operating worldwide with millions of concurrent users
- architectural patterns for reactivity (react in a timely manner), elasticity (react to variable load conditions), resilience (recover from failures)
- instantiate well-known frameworks in projects

Four specialisations

Artificial Intelligence

Machine learning

Game Theory

Bio-informatics

Natural Language Processing

Computational Intelligence

Data Management & Analytics

Data Mining

Big Data Management

Information Retrieval

Human-Data Interaction

Data Visualisation

Software Languages & Software Engineering

Language Design and Implementation

Development Tools and Methods

Program Analysis

Program Evolution

Multimedia

Image Processing

Data Compression

Computer Vision

Speech Synthesis and Coding

- designed for students with solid, fundamental background in CS
- provides deep knowledge of CS in general, and in one of four specialisations

Four specialisations

Artificial Intelligence

Machine learning
Game Theory
Bio-informatics
Natural Language Processing
Computational Intelligence



- study and apply **state of the art** techniques
 - logic-based, statistical, probabilistic, neural inference systems
 - multi-agent systems with emergent intelligence
 - language processing and computational intelligence
- rigorous **mathematical and scientific** context
- supported by **oldest AI group** on the European continent (1982)

Four specialisations



Data Management & Analytics

Data Mining
Big Data Management
Information Retrieval
Human-Data Interaction
Data Visualisation

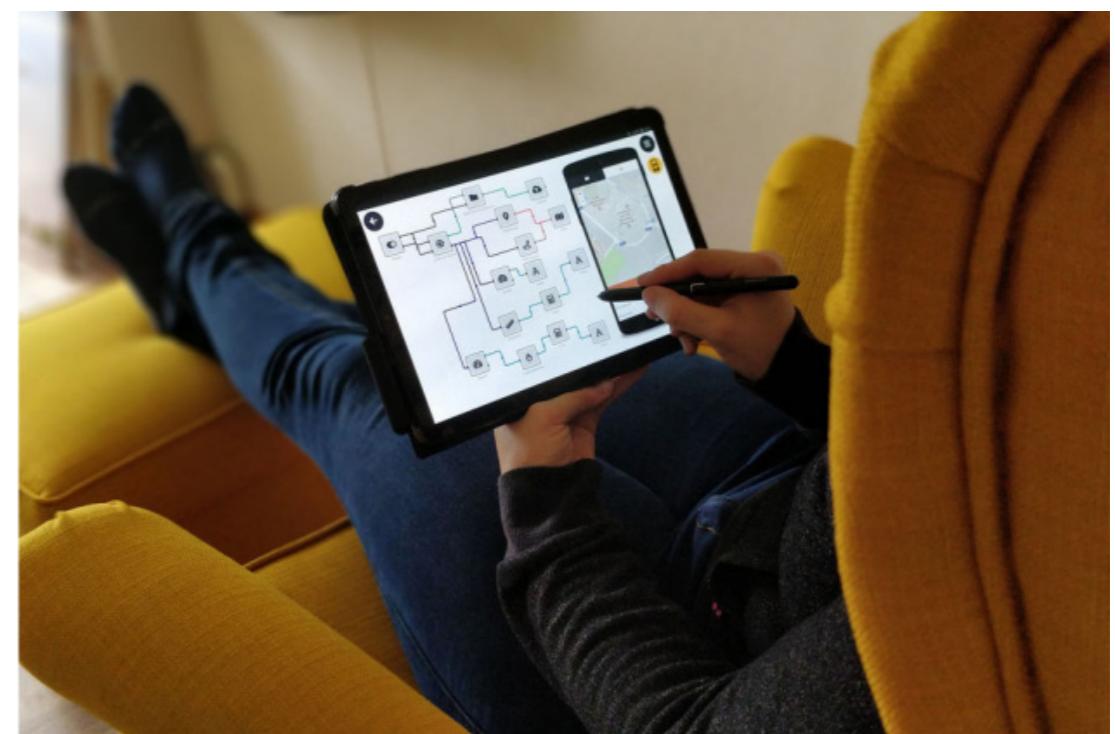
- at the confluence of **Big Data** and **Interactive Data Science**
 - new since 2020-2021!
- study and apply algorithms, techniques, architectures
 - big data management, mining, visualisation, interaction
- supported by all research groups of the department

Four specialisations

- the art, science, and engineering of software construction
 - theoretical foundations: type theory, formal proofs, program analysis
 - language design: paradigms for distribution and concurrency
 - language implementation: compilers and interpreters
 - software engineering: software quality, meta-programming
- supported by a leading research group in PL and SE

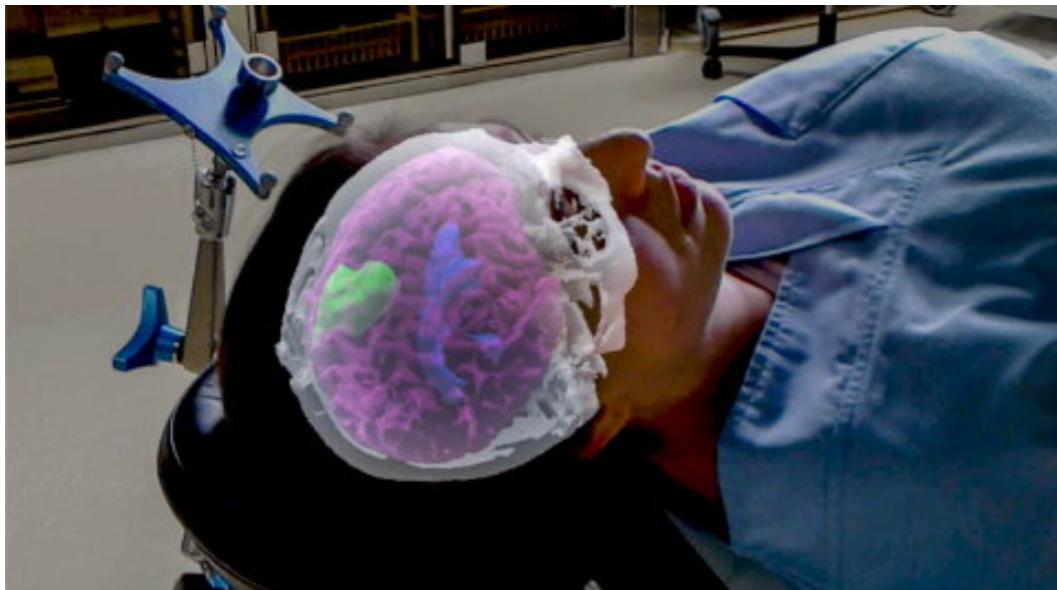
Software Languages & Software Engineering

Language Design and Implementation
Development Tools and Methods
Program Analysis
Program Evolution



Four specialisations

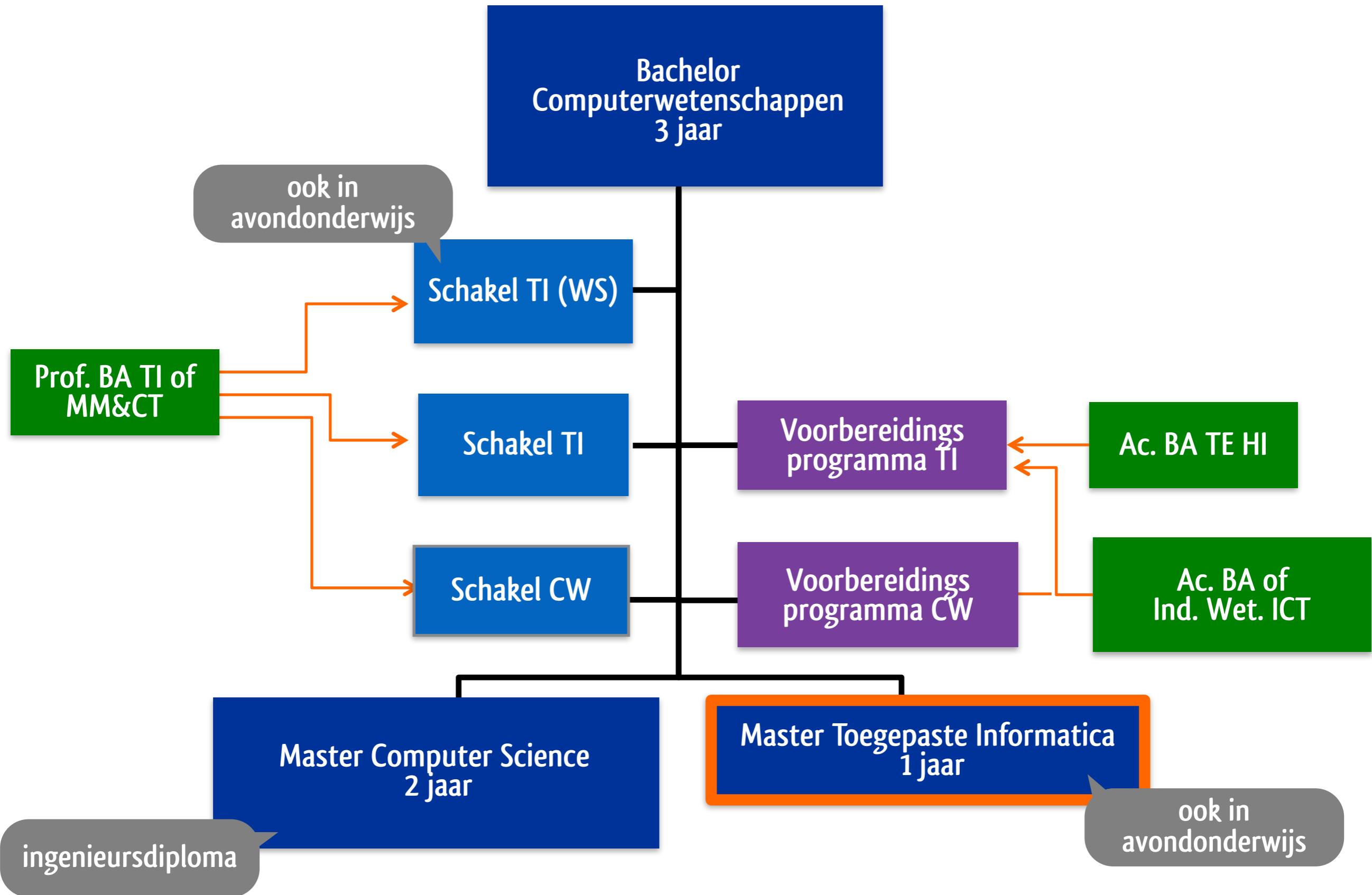
- study **signal processing and transmission** for multimedia content
- supported by a leading research group in electronics and holography



Multimedia

Image Processing
Data Compression
Computer Vision
Speech Synthesis and Coding

Master of Science in Toegepaste Informatica



Programma 1-jarige Master Toegepaste Informatica

48 ECTS verplicht

Methoden van Wetenschappelijk Onderzoek
Schaalbare Datamanagementsystemen
Information Retrieval en Data Mining
Informatievisualisatie
Cloud computing en Big Data Processing
Statistical Foundations of Machine Learning
Masterproef (15 ECTS)



12 ECTS keuzevakken

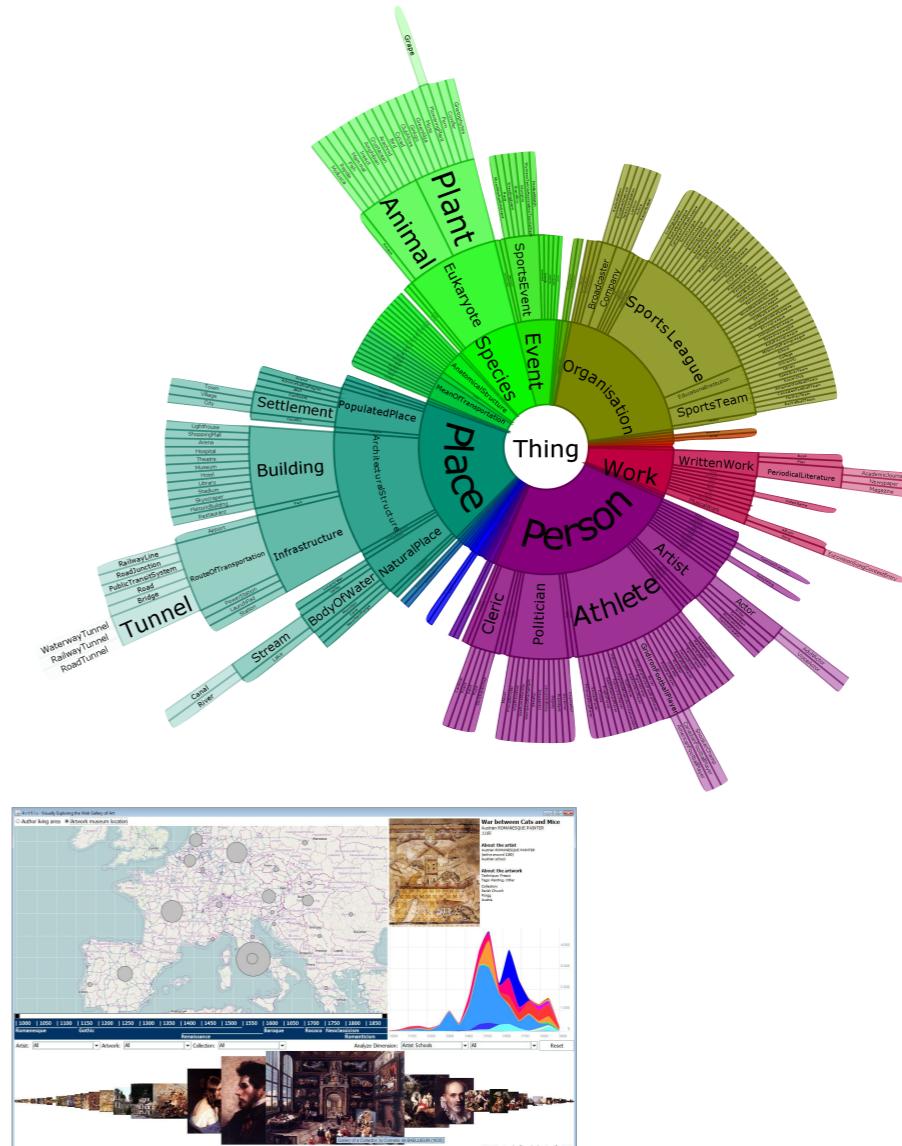
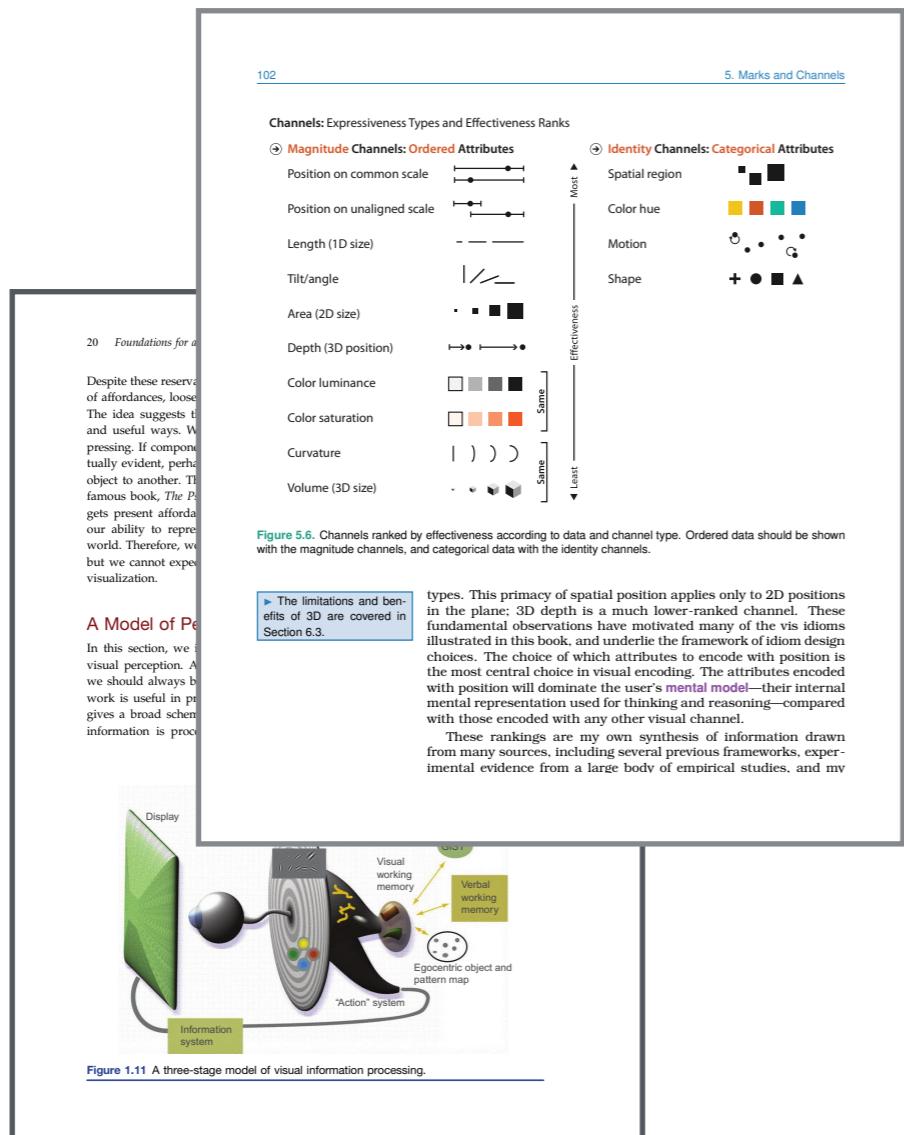
Software Architecturen
Security in Computing
Natural Language Processing
Declaratief Programmeren

(+ mits goedkeuring: stage, reparativvakken)

Informatica-aspecten
van Data Science en
Big Data toepassingen

- na academische bachelor of schakel -en voorbereidingsprogramma
- meer toepassingsgericht dan 2-jarige master
- theoretische basis met mogelijkheid om kennis in de praktijk toe te passen

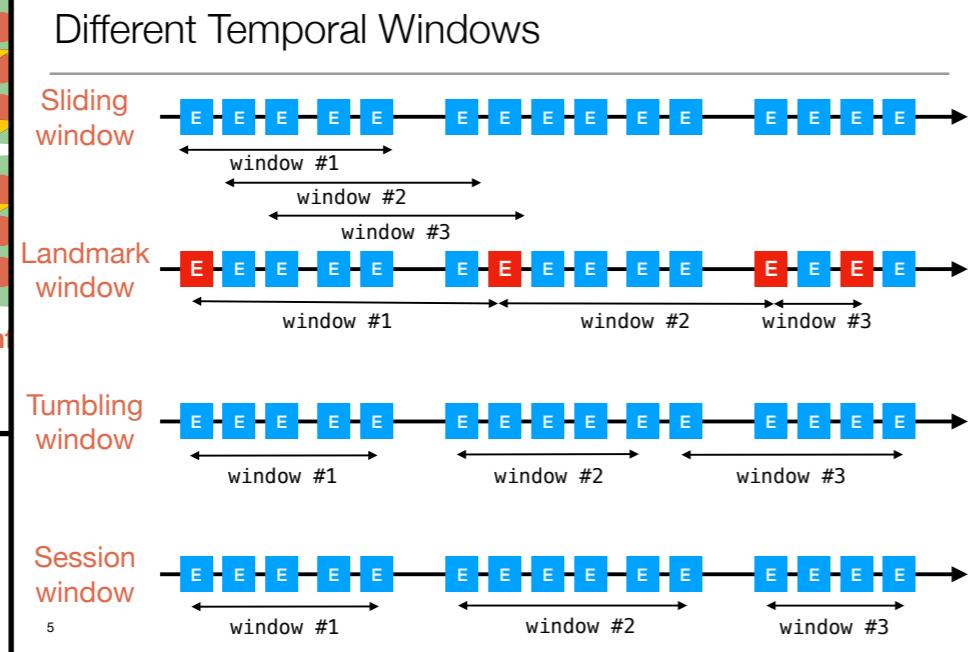
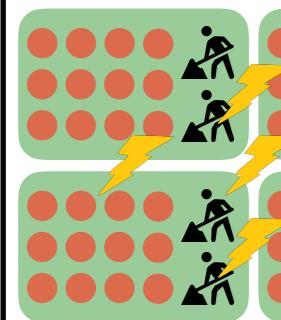
Voorbeeldvak: Informatievisualisatie



- visualisatietechnieken voor exploratieve analyse van grote datasets
- case studies illustreren perceptietheorie en interactietechnieken
- in groepsverband visualisatieoplossing ontwikkelen

Voorbeeldvak: Cloud Computing & Big Data Processing

From Data-Parallel to Distributed Data-Parallel



3

5

Proof of work for block #n

(often) useless

make adding a block artificially difficult

Bitcoin mining stats

<https://bitcoinwisdom.com/bitcoin/difficulty>

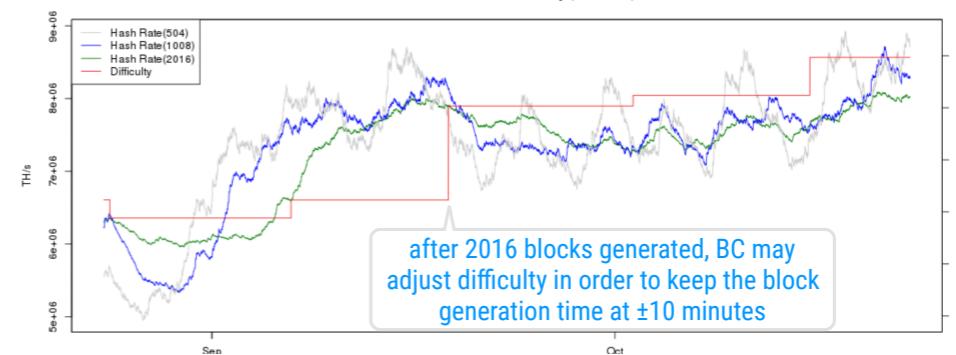
hash upper bound

Bitcoin Difficulty: 1,196,792,694,099
Estimated Next Difficulty: 1,163,876,864,166 (-2.75%)
Adjust time: After 742 Blocks, About 5.3 days
Hashrate(2): 8,736,302,324 GH/s
Block Generation Time(2): 1 block: 10.3 minutes
3 blocks: 30.9 minutes
6 blocks: 1.0 hours
Updated: 17:0 (2.2 minutes ago)

calculator: hashes/s → BC earnings

Difficulty	KH/s	MH/s	GH/s	TH/s	BTC/hour	BTC/day	BTC/week	BTC/month	USD/hour	USD/day	USD/week	USD/month
1000000	8.755e-9	2.181e-7	0.00001471	0.000006383	0.0002229	0.0005349	0.003745	0.01685				
1000	8.755e-9	2.181e-7	0.00001471	0.000006383	0.0002229	0.0005349	0.003745	0.01685				
1	8.755e-9	2.181e-7	0.00001471	0.000006383	0.0002229	0.0005349	0.003745	0.01685				
0.001	8.755e-9	2.181e-7	0.00001471	0.000006383	0.0002229	0.0005349	0.003745	0.01685				

Bitcoin Hash Rate vs Difficulty (2 Months)



...
ly less
10
utes

s from
locks"

25

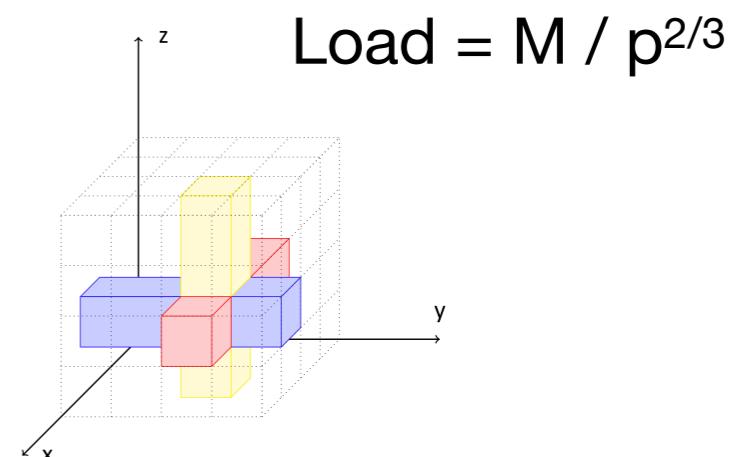
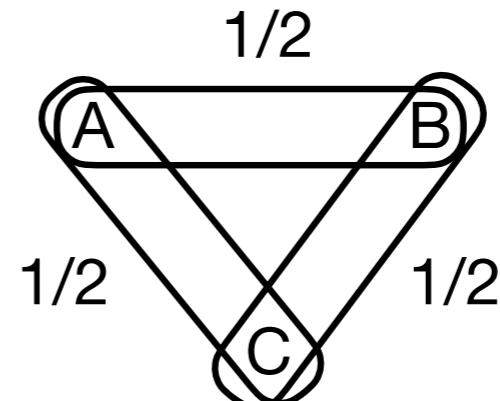
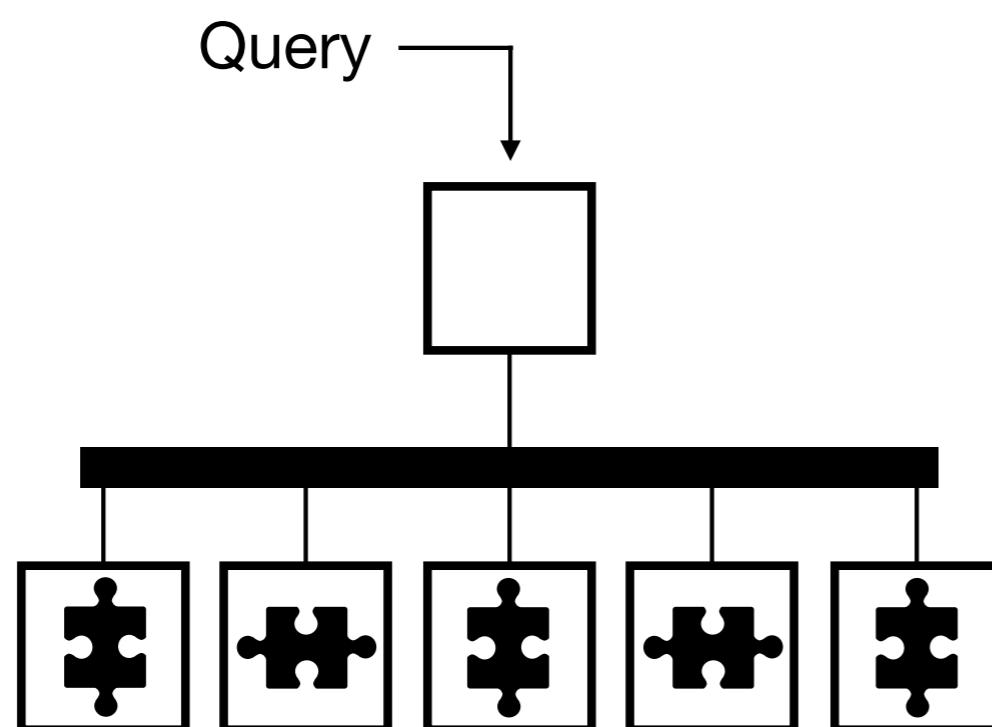
26

- cloud computing: decentralisatie (Blockchain), consistentie, beveiliging, ...
- big data processing: cluster computing, gegevenslocaliteit, streaming (Spark)
- raamwerken illustreren fundamentele concepten van beide domeinen

Voorbeeldvak: Schaalbare Datamanagementsystemen

What is the “best” algorithm to compute a query?

$$R(A,B) \bowtie S(B,C) \bowtie T(C,A)$$



- gedistribueerde databanken voor beheer van grote datasets
- principes onderliggend aan systemen zoals MapReduce
- uitdagingen van partitionering, query planning, transactieverwerking