STRATEGIEËN VOOR HET MODELLEREN VAN HTML5-APPLICATIES

WIE BEN IK?

Niels Bergsma

Software engineer bij Reflex Online

Modelleer- en UX-fanaat

DEZE PRESENTATIE

Gaat over bewustwording & beslissingen

Niet over een specifieke technologie

WAAROM?

Trend

Nieuwe mogelijkheden HTML5

Wildgroei frameworks / libraries

Toenemende eisen / wensen eindgebruikers

Bevorderd polyglot ontwikkeling

HEEL VROEGER

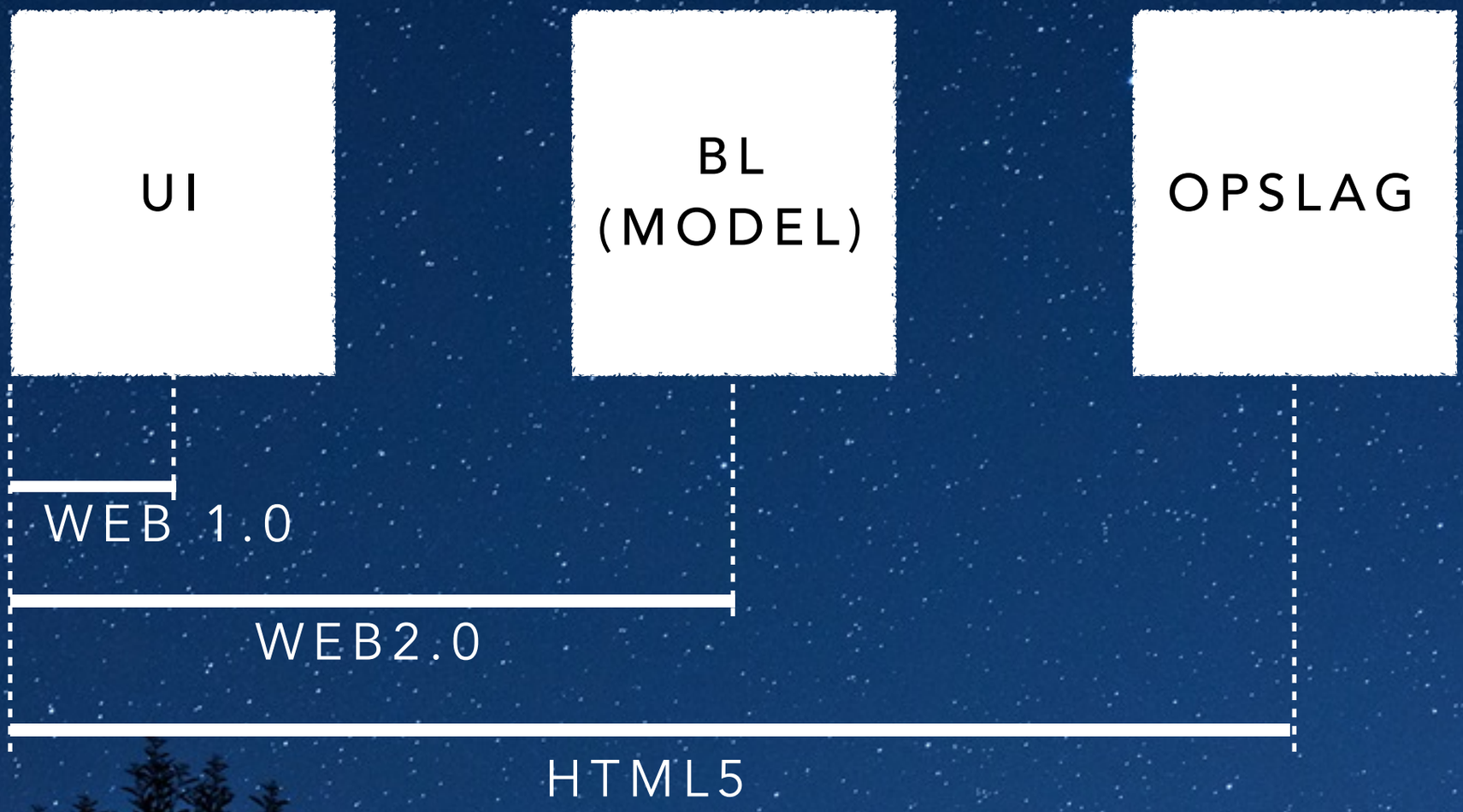
VROEGER


CLIENT
GET /PRODUCT/ABC
PAGINA (HTML)
INFO (XML)
REVIEWS (XML)
SERVER

NU


CLIENT
SERVER
GET /PRODUCTS
PAGINA (STATIC HTML)
GET /PRODUCTS/ABC
INFO (JSON)
REVIEWS (JSON)

CLIENT VERANTWOORDELIJKHEDEN



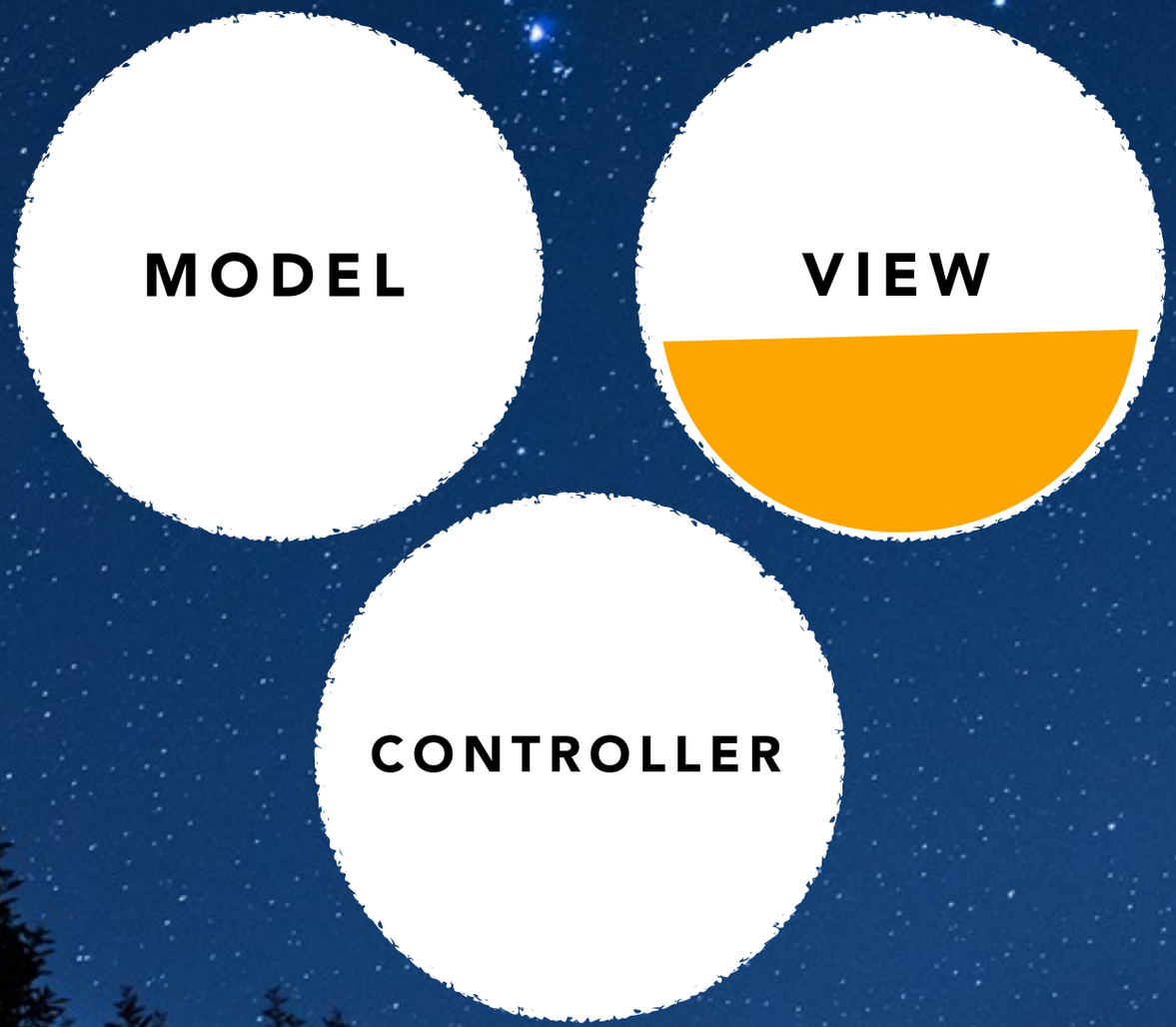
UI UITGELICHT

MODEL

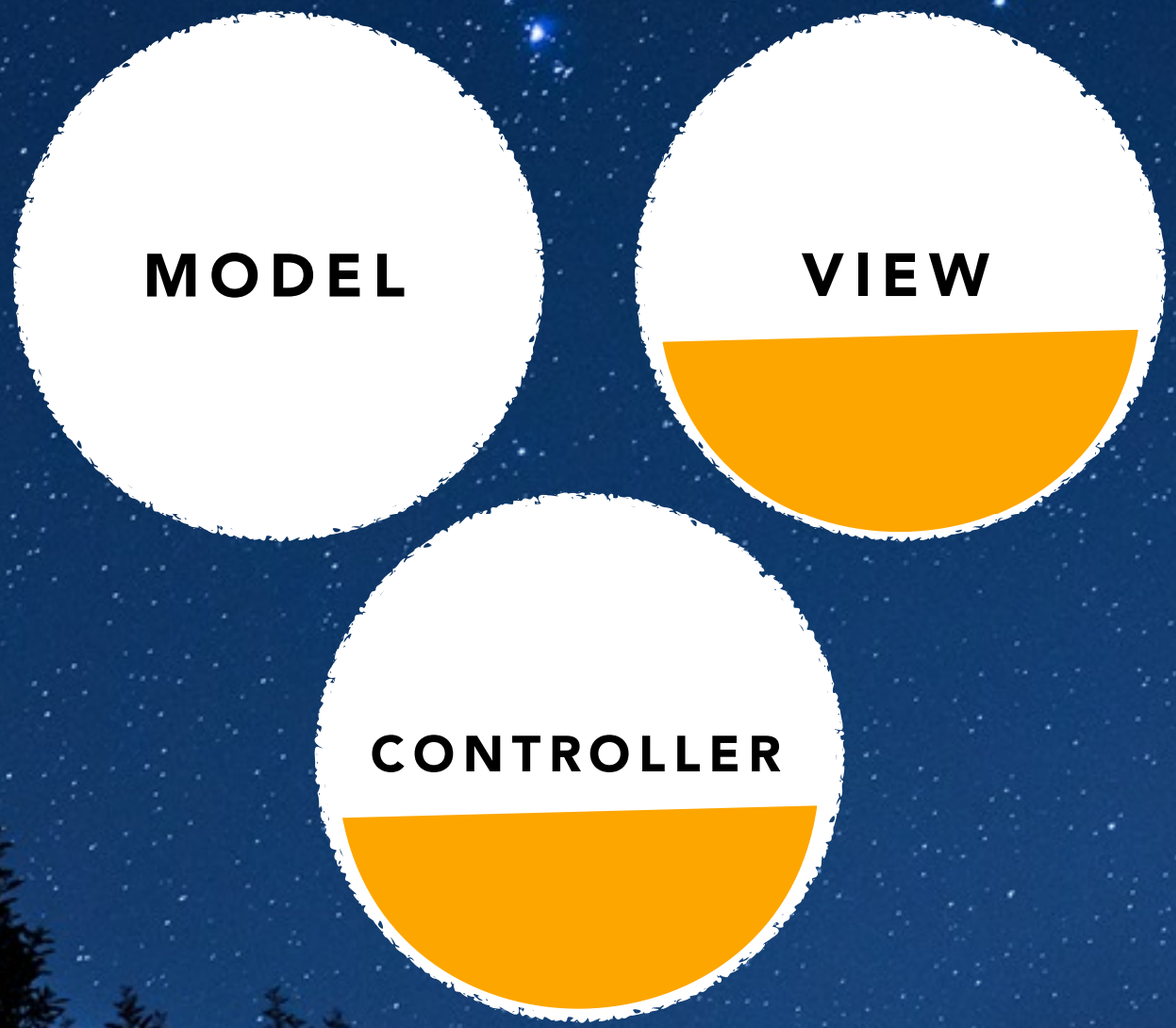
VIEW

CONTROLLER

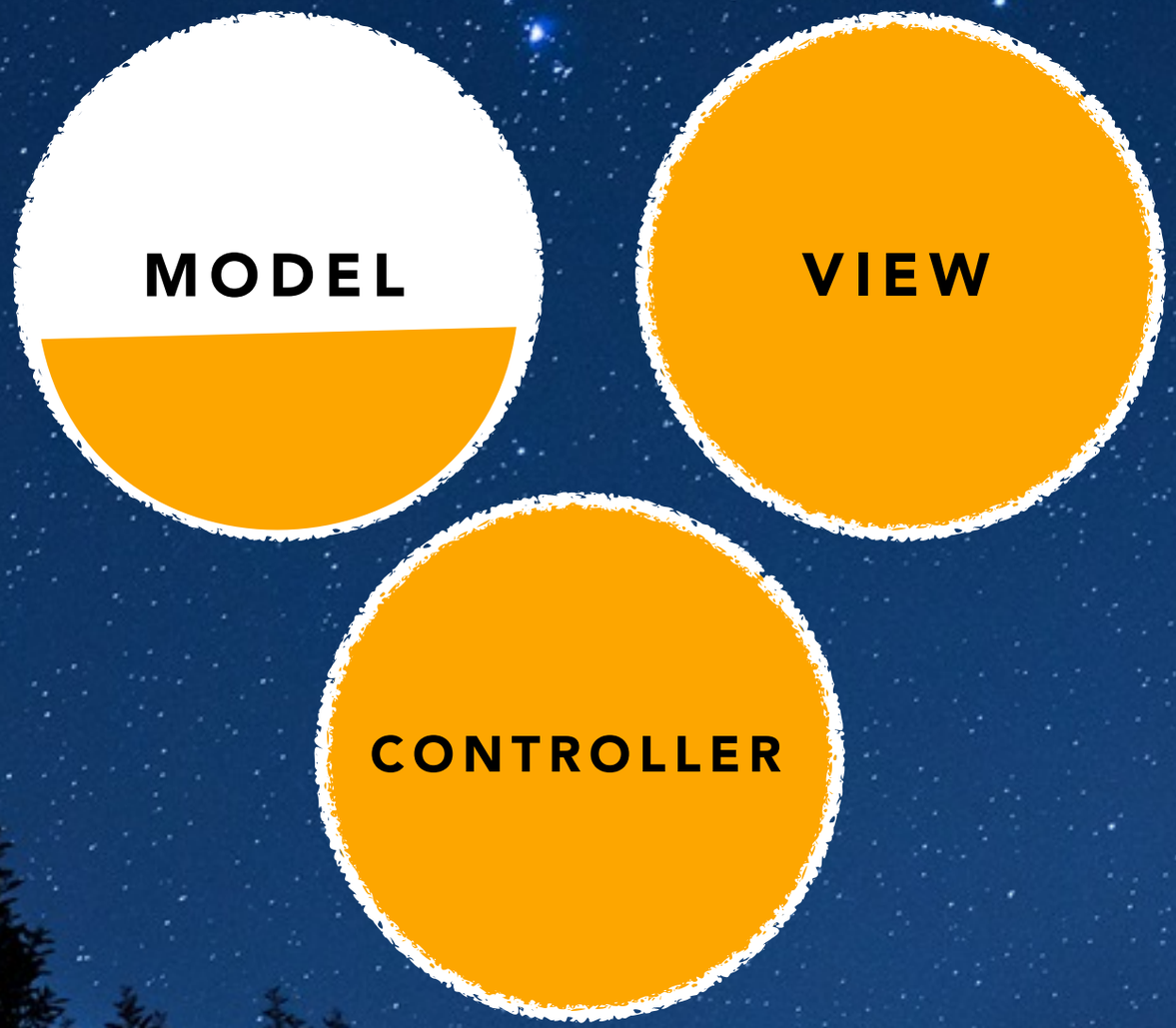
UI UITGELICHT (WEB 1.0)



UI UITGELICHT (WEB 2.0)



UI UITGELICHT (HTML5)



HTML5?

Storage (KV / SQL)

Offline cache

Websockets

Webworkers

WAAROM?

UX

Schaalbaarheid (cache, rendering, pay load)

Forceert scheiding UI - model / client - model

Opdelen ontwikkeling (cycles)

API direct beschikbaar

IS DIT NIEUW?

Nee

Maar andere technologie stack

Het gaat om bewustwording

AFWEGING

UI

BL
(MODEL)

OPSLAG

Data (API) call karakteristiek

←Chatty (RPC) Hogere payload →

AFWEGING

UI

BL
(MODEL)

OPSLAG

Initiële laden (cold start)

Grootte (MB) →

AFWEGING

UI

BL
(MODEL)

OPSLAG

Cacheability

Beter →

AFWEGING

UI

BL
(MODEL)

OPSLAG

Code duplicatie

Meer →

AFWEGING

UI

BL
(MODEL)

OPSLAG

Beveiliging

Lastiger →

AFWEGING

UI

BL
(MODEL)

OPSLAG

SEO

Lastiger →

AFWEGING - SAMENVATTING

Omgeving

Gebruikers / bezoekgedrag

Data + gevoeligheid

SEO

TYPISCHE ARCHITECTUUR?

MVW (MVC / MVVM / MVP)
Component based vs module based
Single page vs multiple pages
Declaratief vs imperatief
Angular vs Knockout vs Ember vs…
…

3-tier
Hexagonal (P&A)
Actor
Micro services +(distributed)
CQRS
(NO)SQL
….

Client

Server

STRUCTUUR

HTML+
PARTIALS

(UI)

JS
MODULES

(BL)

LOCAL
STORAGE

Client

STRUCTUUR

UI

BL

OPSLAG

Client

BL

OPSLAG

Server

VOORBEELD

Bootstrap (CSS)
AngularJS (MVW)
LocalForage
no jQuery *

Client

Server

SAMENVATTING & TIPS

1 strategie

Horizontaal / verticaal opdelen

Minimaliseer dependencies

Trust no one

DISCUSSIE

Progressive enhancement

- minimum bepalen
- zonder trucs werkend maken
- verreiken met nieuwe mogelijkheden

Graceful degradation

- minimum bepalen
- werkend maken met nieuwe mogelijkheden
- gaten opvullen met polyfills

DISCUSSIE

Monolithic applicatie

- 1 deployable unit

Micro services / SOA

- meerdere deployable units

DISCUSSIE

Stelling: thick clients trend is tegenstrijdig met de cloud trend

VRAGEN

BEDANKT

Twitter: @NielsBergsmaNL

GitHub: NielsBergsma

LinkedIn: Niels Bergsma