

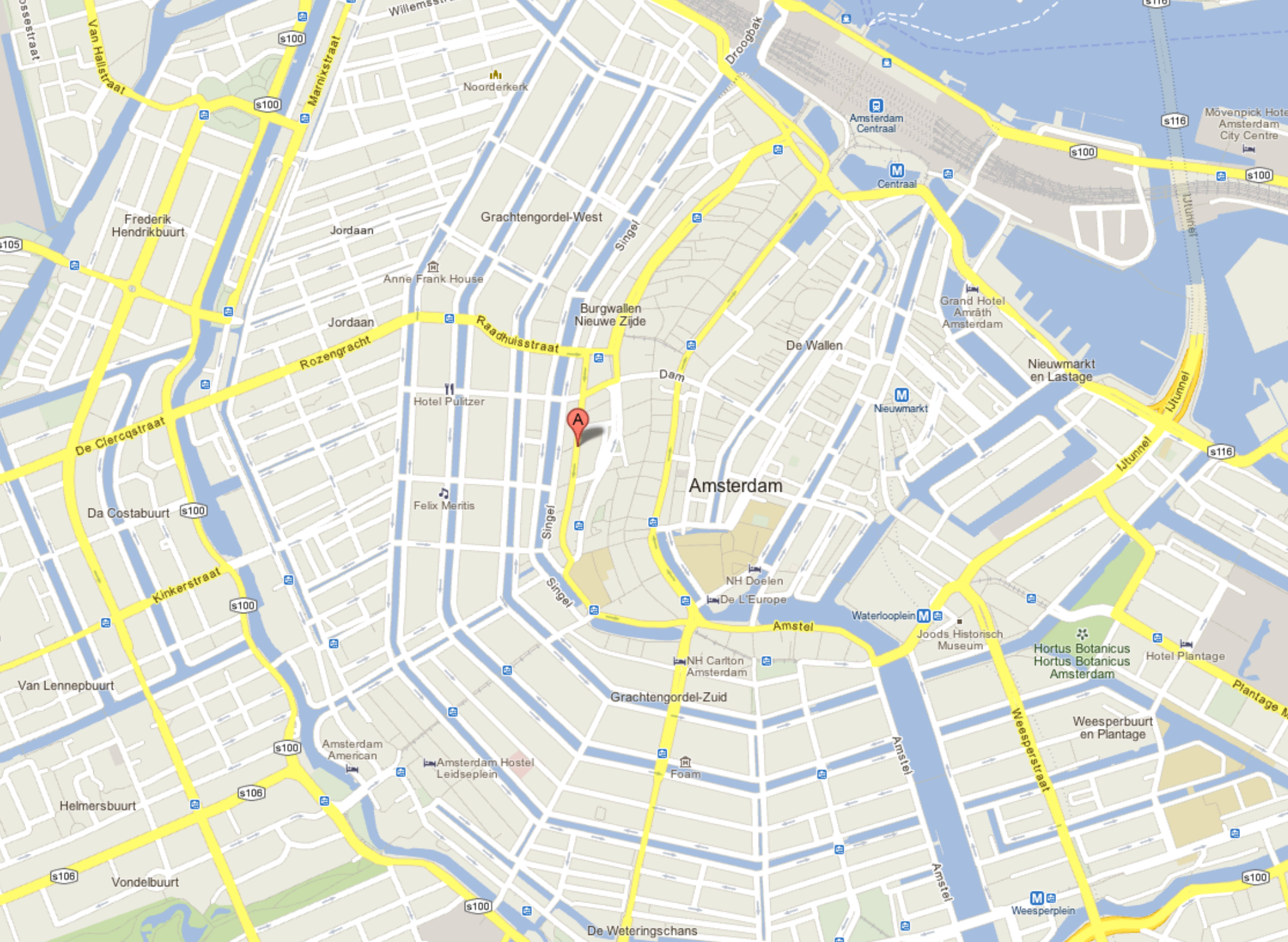
# Functional Programming at

**Silk** 

Sebastian Visser - 2012

# Company

- Internet startup
- Working on one product
- Started 3 years ago
- Grown from 4 to 11 people
- Seed funding by American VC
- Located in Amsterdam



# Amsterdam



Noorderkerk

Amsterdam Centraal

Frederik Hendrikbuurt

Jordaan

Grachtengordel-West

Anne Frank House

Burgwallen Nieuwe Zijde

Centraal

Grand Hotel Amr ath Amsterdam

Jordaan

Rozengracht

Raadhuisstraat

De Wallen

Nieuwmarkt en Lastage

Dam

Hotel Pulitzer

Nieuwmarkt

De Clercqstraat

Da Costabuurt

Felix Meritis

NH Doelen

De L'Europe

Waterlooplein

Joods Historisch Museum

Hortus Botanicus Amsterdam

Hotel Plantage

Van Lennepbuurt

NH Carlton Amsterdam

Grachtengordel-Zuid

Amsterdam American

Amsterdam Leidseplein

Foam

Weesperbuurt en Plantage

Helmersbuurt

Vondelbuurt

De Weteringschans

Weesperplein

# Product

- App to manage structured content
- Sites are collections of pages
- MediaWiki meets Google documents

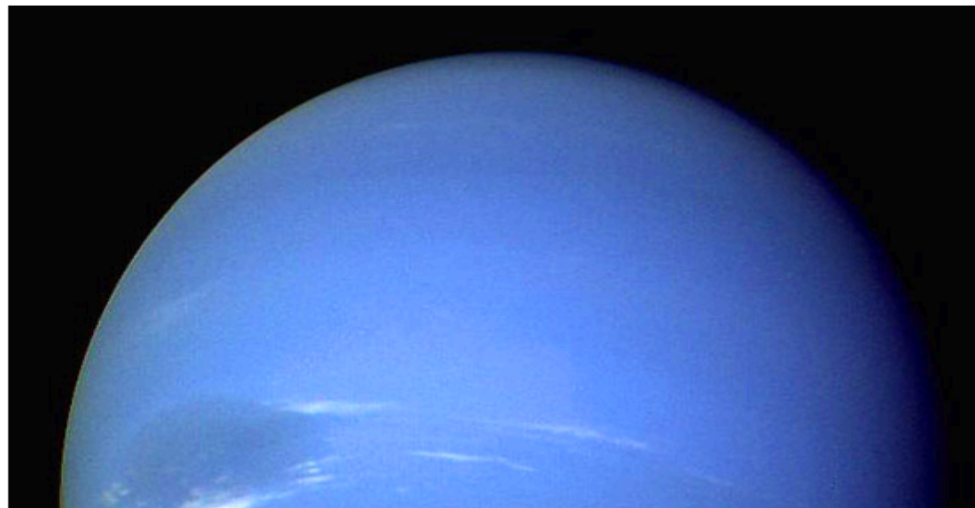
# Semantic web

- Facts in pages can be tagged
- **Tags** form a **semantic graph**
- The graph can be explored

Home > Planet > Neptune

## Neptune

Neptune is the **eighth** and farthest planet from the Sun in the Solar System. Named for the Roman god of the sea, it is the fourth-largest planet by diameter and the third largest by mass. Neptune is **17 times the mass of Earth** and is slightly more massive than its near-twin Uranus, which is 15 times the mass of Earth but not as dense. On average, Neptune orbits the Sun at a distance of 30.1 AU, approximately 30 times the Earth–Sun distance. Its astronomical symbol is ♆, a stylized version of the god Neptune's trident.



Page saved

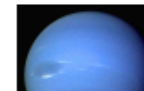
Exit Edit Mode

Change page settings

### Planet

Planet: **eighth**

Image:



Diameter: **51118 km »**

Distance to the Sun: **19.22941195 AU »**

Body Type: **gassy**

Kind: **gas giant**

Orbital Period: **30799.095 days**

01 · VISUALIZATION

Moon · Planet · Star



options ▶

02 · TAGS

Display a tag ▼

Planet Refine Sort By

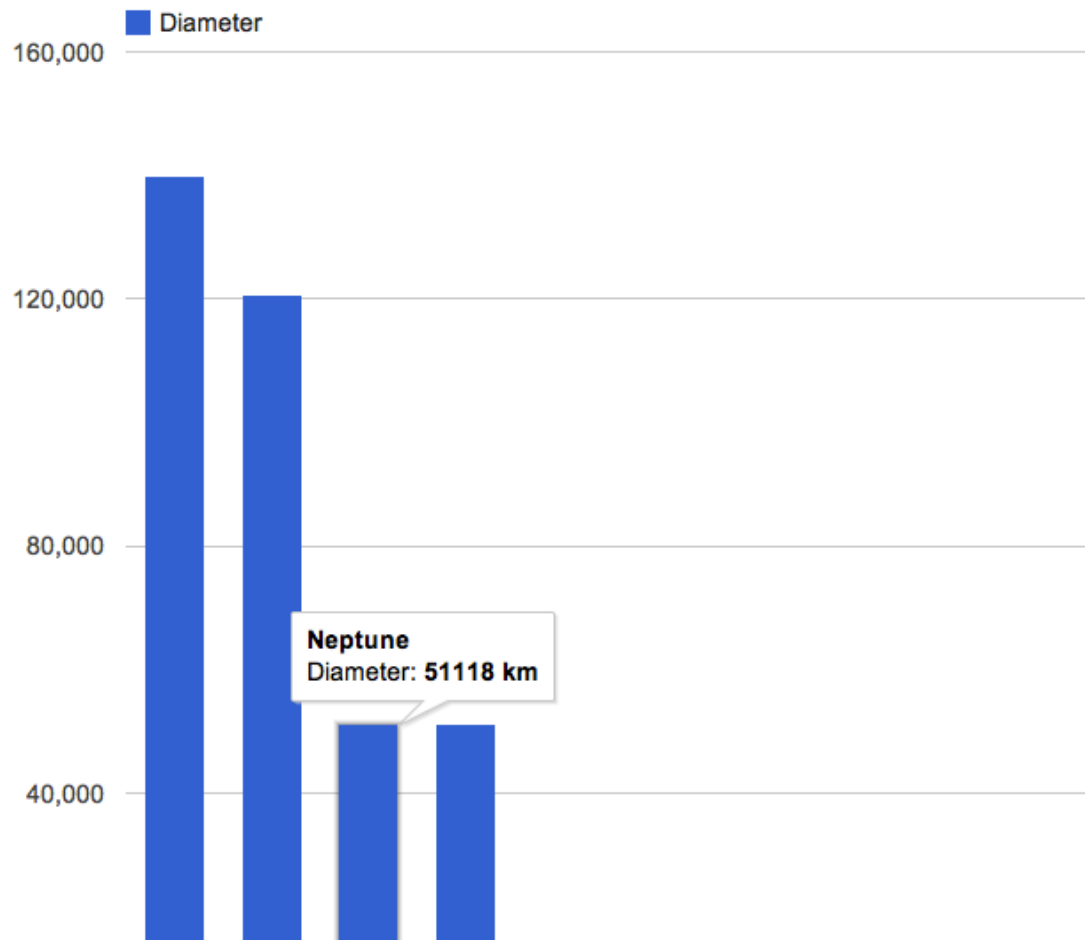
Diameter Refine ↓ ⊗

Image Refine Sort By ⊗

Home > Column chart of the Diameter for Planet pages



# Column chart of the Diameter for Planet pages



# Why am I here?

Our servers and tools  
are all written in **Haskell**.



# Client

Entirely written in **JavaScript**.  
This is a challenge.

Let's not talk about that...

# Service oriented

- An **API** server implementing domain logic
- Two **front-end** servers (website/product)
- A geo-location **enrichment queue**
- A variable number of **query servers**
- A **pool manager** for the query servers
- A **status panel** to monitor our cloud
- A **mailer** process for status updates

# HTTP/1.1 200 OK

All servers talk HTTP,  
either using **Happstack** or **Snap**.

# RESTful API

From **one definition** we **generate**

- Happstack/Snap handlers
- Haskell client
- JavaScript client
- Ruby client
- API documentation

# Templating

Front-end servers use

**Heist** for templates,  
**Loot** for routing.

# Databases

We use quite a few of them.

- **PostgreSQL** for domain objects
- **filesystem** for Silk pages
- **in-memory Haskell** semantic index
- **MongoDB** as semantic cache
- **memcached** for query results

# Advantages of Haskell

- Strong types prevent bugs
- Refactoring is a bliss
- Active community
- Lots of packages
- Package management
- Seriously great compiler

# Refactoring

1. Change a bit of code
2. Compile your package
3. Fix type errors
4. Repeat with 2 for all dependencies
5. **Everything works!**



# newtype

```
type Site = Text
```



```
newtype Site = Site Text  
make = Site . normalize  
asText (Site site) = site
```

But...

# Disadvantages

- Overall library quality is disappointing
- Reasoning about resources is hard
- Build process can be overwhelming

# Library quality

- Too many libs solving the same problem
- Lack of documentation
- Overly complicated interfaces
- Bad performance
- Outdated dependencies

# XML mess

- HaXml
- libxml
- Haskell XML Toolkit (HXT)
- XML Light
- XML conduit
- xmlhtml
- tagsoup

# String mess

- `Data.String`
- `Data.Text`
- `Data.Text.Lazy`
- `Data.ByteString`
- `Data.ByteString.Lazy`

Every package has its own preference.

# API design

- Excessive use of type classes
- Complicated monad transformer stacks
- Exposing the abstract internals by default
- Bad naming of functions
- No structure in documentation
- Not exposing crucial parts the library

# Resource spikes

What to do when

- a servers becomes unresponsive?
- CPU usage goes up to 600%?
- memory goes up from 1GB to 4GB?



# Laziness

Types improve reasoning over code.

Laziness worsens reasoning over  
resource utilization.

# Rule of thumb

**Strictly write** values to memory.

**Lazily read** values from memory.

# Dependency Hell

Managing versions is cumbersome.

**Package Versioning Policy** (PVP)

Helps, but has drawbacks.

People sometimes skip upper bounds.

cabal update is awfully slow

# Solutions

- Improved **solver** in cabal-install  $\geq 0.14$
- **Cabal-dev** for isolated builds
- **Bumper** tool for PVP compliant bumps

Still,

if it compiles on my machine,  
it might fail on yours.

# It's not that bad!

Slowly but steadily improving

The most fun language

Problems aren't language related

Other languages have similar problems

# It's not that bad!

Slowly but steadily improving

The most fun language

Problems aren't language related

Other languages have **worse** problems

# Silk is hiring!

Development Operations

Front-end engineers

Currently no Haskell position



[silkapp.com](http://silkapp.com)  
[github.com/silkapp](https://github.com/silkapp)



@sfvisser  
+Sebastian Visser  
[github.com/sebastianvisser](https://github.com/sebastianvisser)