

*Langages
& Modèles
Dédicés*

Déroulement

6×4h – cours + TP + mise en commun
lundi/jeudi après-midi

3 sujets de TP – ×2 séances
en groupes (4-5)

3 notes – présentation ou compte-rendu
par groupe, en fin de 2e séance

The Addison-Wesley Signature Series

DOMAIN-SPECIFIC LANGUAGES

MARTIN FOWLER
WITH REBECCA PARSONS



A MARTIN FOWLER SIGNATURE Book
Martin Fowler

Martin Fowler
martinfowler.com/books/dsl.html
catalog of DSL patterns



DSL Engineering

*Designing, Implementing and Using
Domain-Specific Languages*

Markus Voelter

with Sebastian Benz, Christian Dietrich, Birgit Engelmann
Mats Helander, Lennart Kats, Eelco Visser, Guido Wachsmuth

dslbook.org

Markus Voelter

disponible en ligne:
<http://dslbook.org>

Language?

the ***mental faculty*** [...] to learn languages
and to produce and understand utterances

a ***formal system of signs***
governed by grammatical rules [...]
to communicate meaning

a ***tool*** for communication



avant Java 8...

```
double minutesToListen = 0;  
Iterator<Song> it = songs.iterator();  
while (it.hasNext()) {  
    Song each = it.next();  
    if (each.played() <= 1)  
        minutesToListen += each.duration();  
}
```

```
double minutesToListen =  
songs.stream()  
    .filter(s -> s.played() <= 1)  
    .mapToFloat(Song::duration)  
    .sum();
```



java.util.stream

```
require 'sinatra'  
  
get '/hello/:name' do  
  "hello, #{params[:name]}"  
end
```



Ruby & Sinatra



Modèle ?

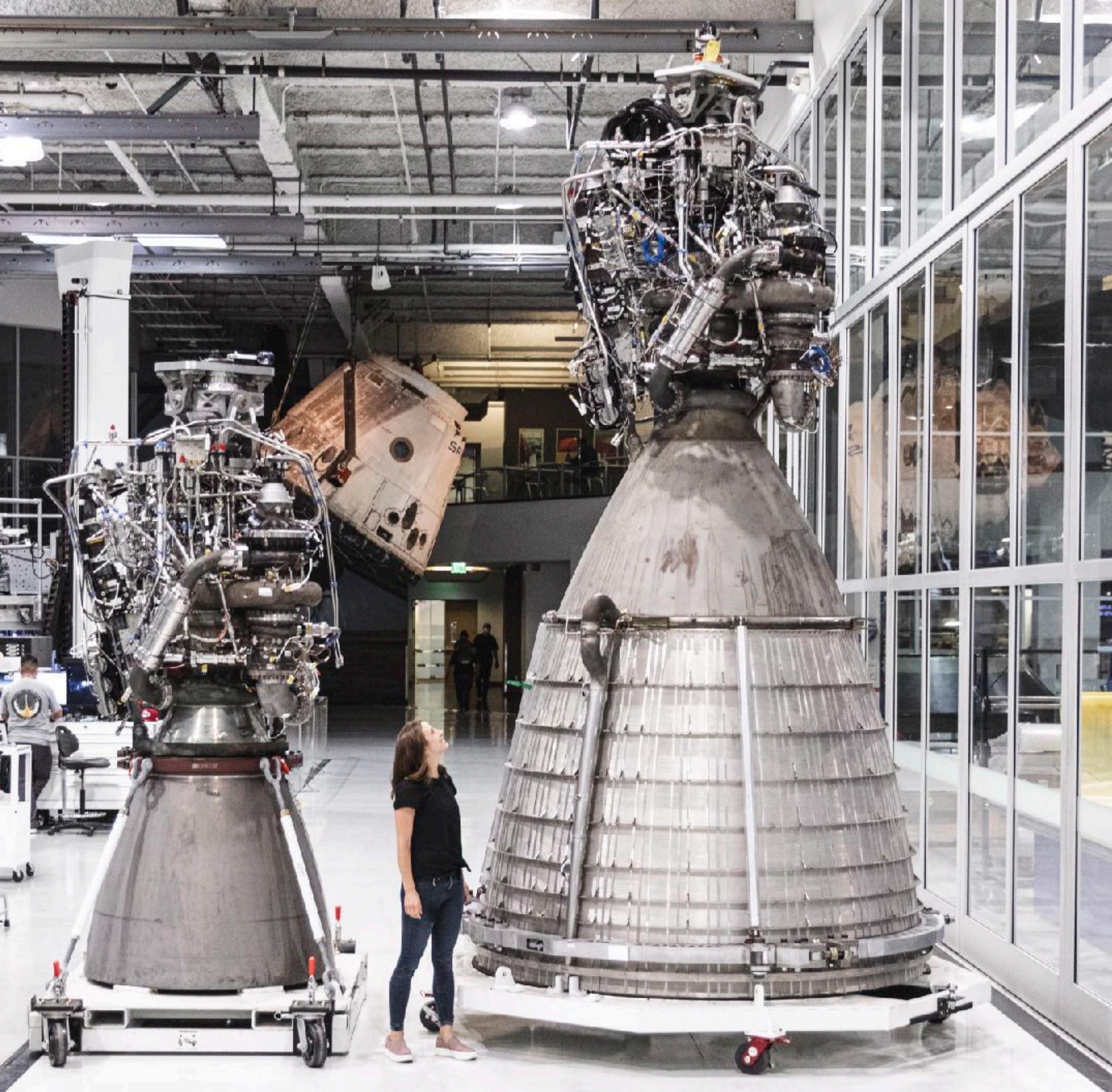
une ***représentation simplifiée*** [· · ·]

de la réalité d'un phénomène

limité à un ***domaine d'application*** [· · ·]

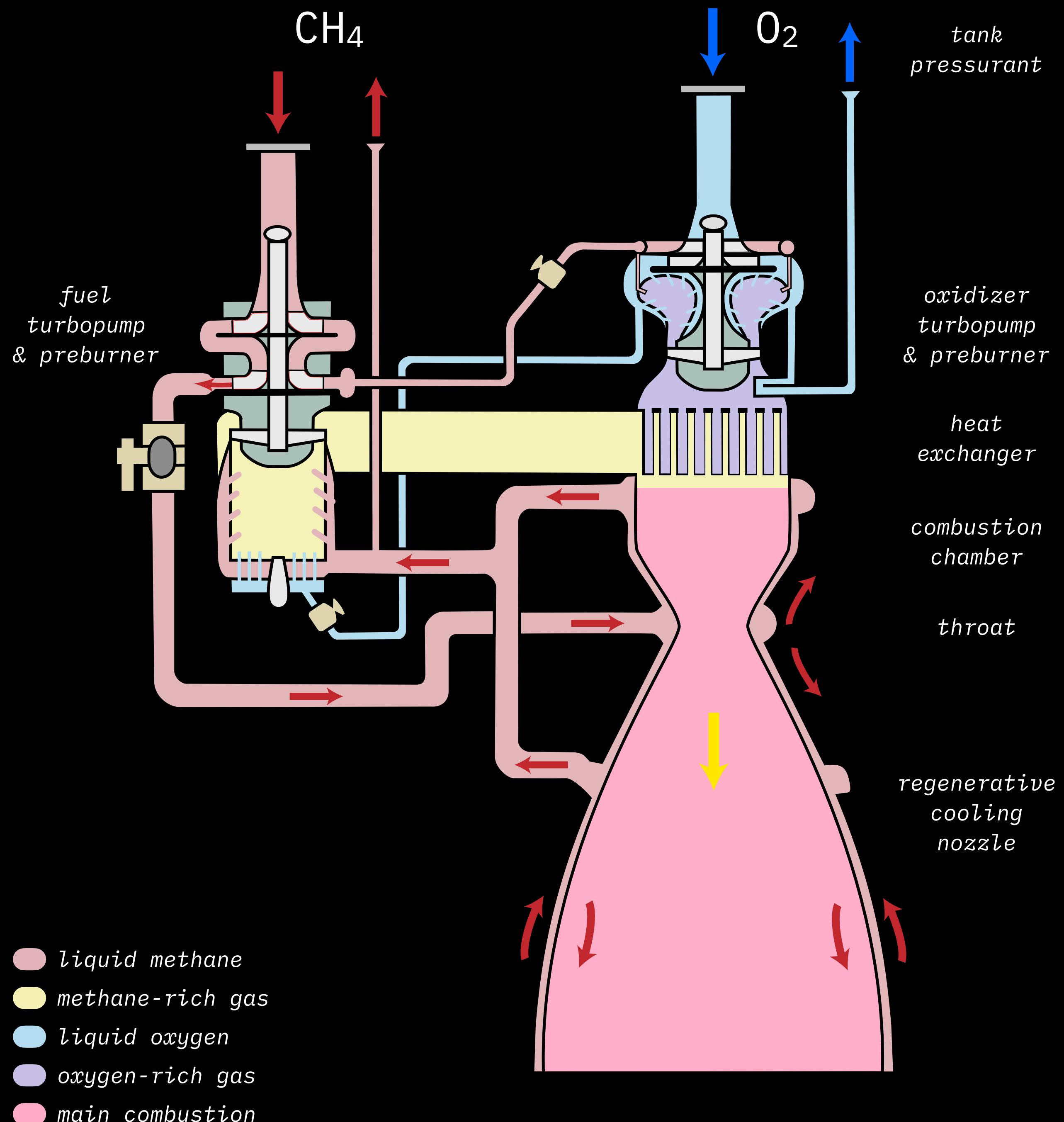
en dehors duquel ledit modèle

n'est plus applicable.



moteurs Raptor
(sea-level & vacuum)

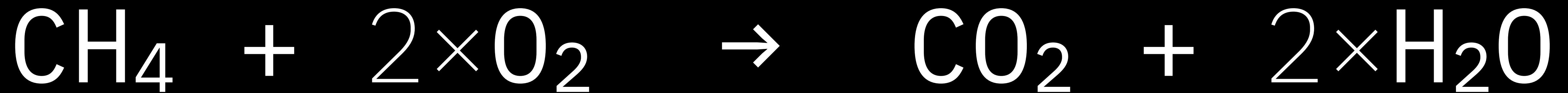
(engineer, for scale)



Raptor
SpaceX, 2022?

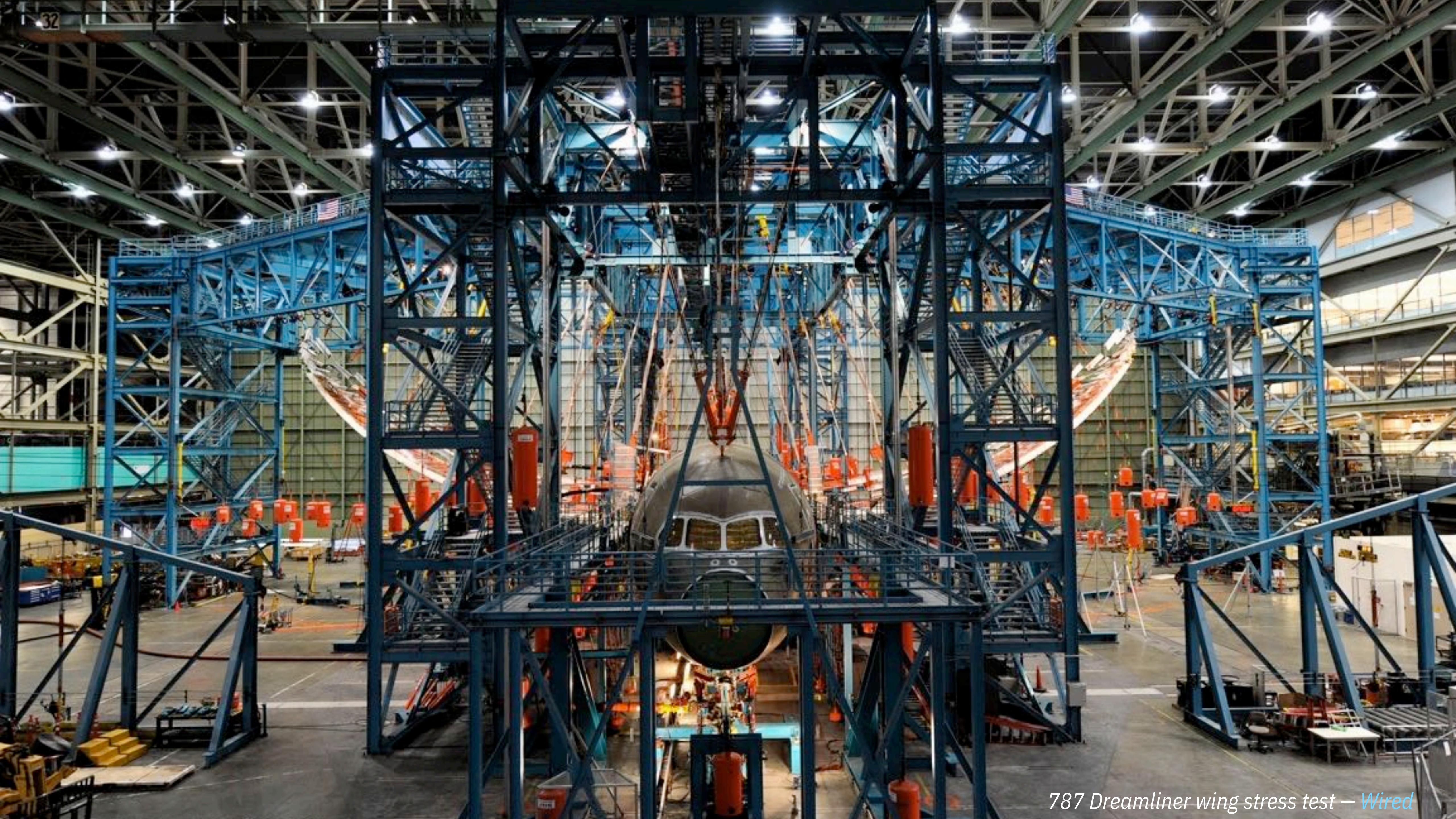
staged combustion
(full-flow)

env. 1500kg – 11Mhp



Chimie (collège)

combustion du méthane



787 Dreamliner wing stress test – *Wired*

ESRI
DEVELOPER SUMMIT
PLENARY
2019

Keynote:
Real Software
Engineering

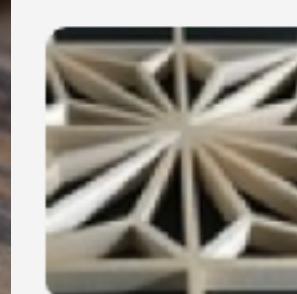
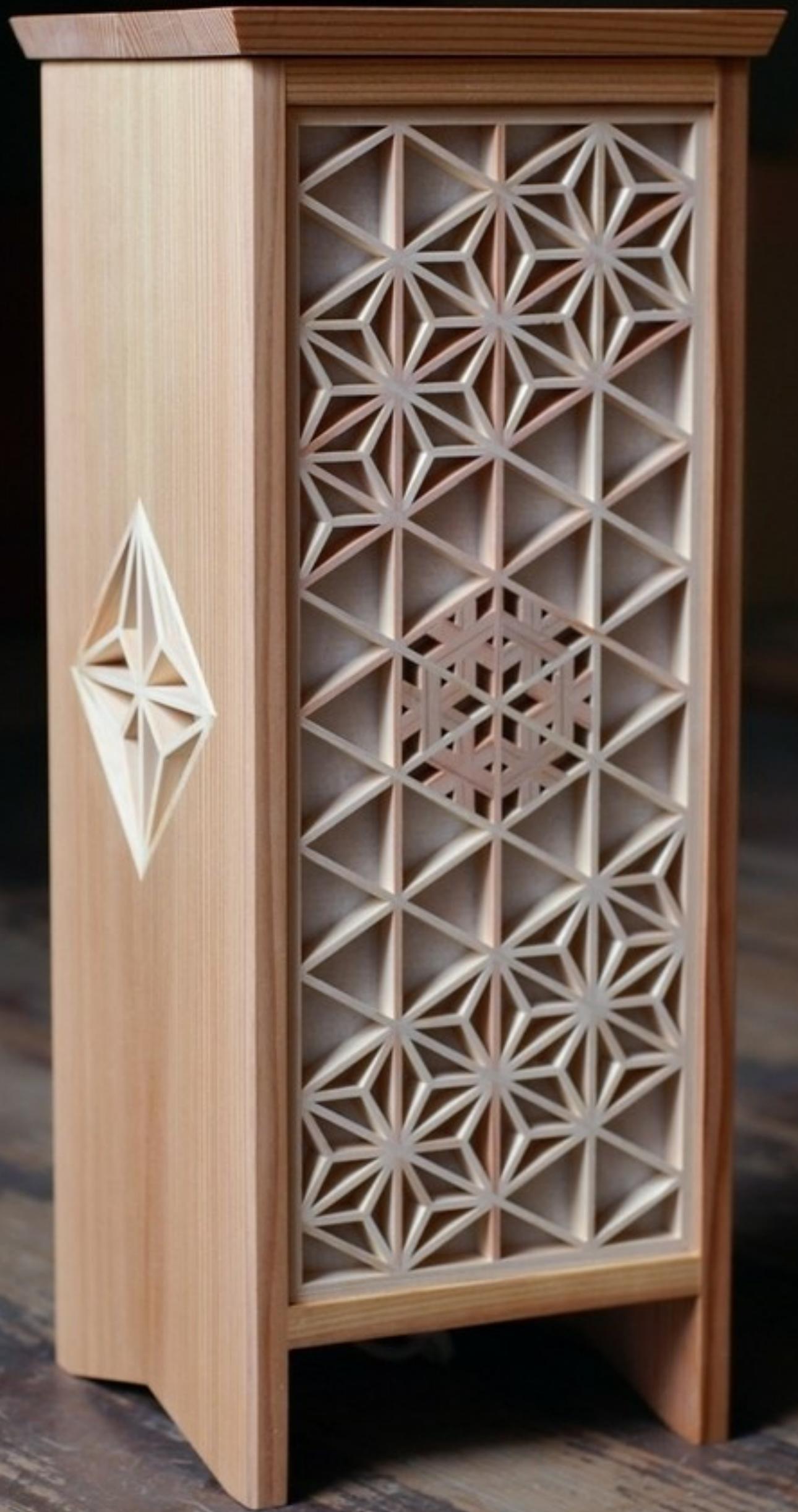


Glenn Vanderburg

Dédié ?

à quel **but**?

- portabilité?
- génération de code?
- analyse, vérification?
- intégration des experts?



Gabarits d'angle pour kumiko

par Séverine il y a 2 ans

Abonné 139

L'Air du Bois

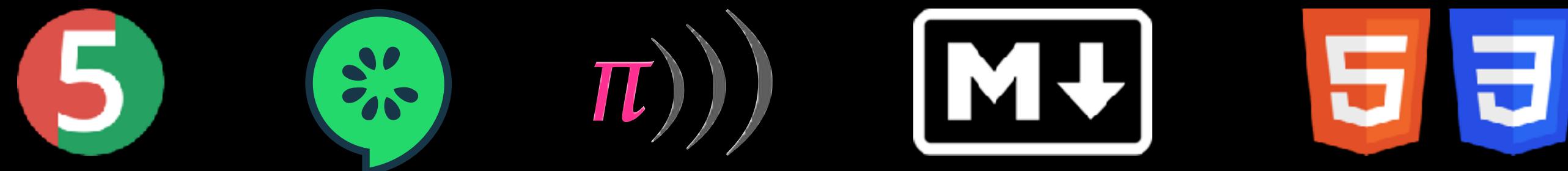
langage à usage général

GPL – general-purpose programming language



langage dédié

DSL – domain-specific language



```
printf("note de %s: %.2d/20", nom, moy);
```

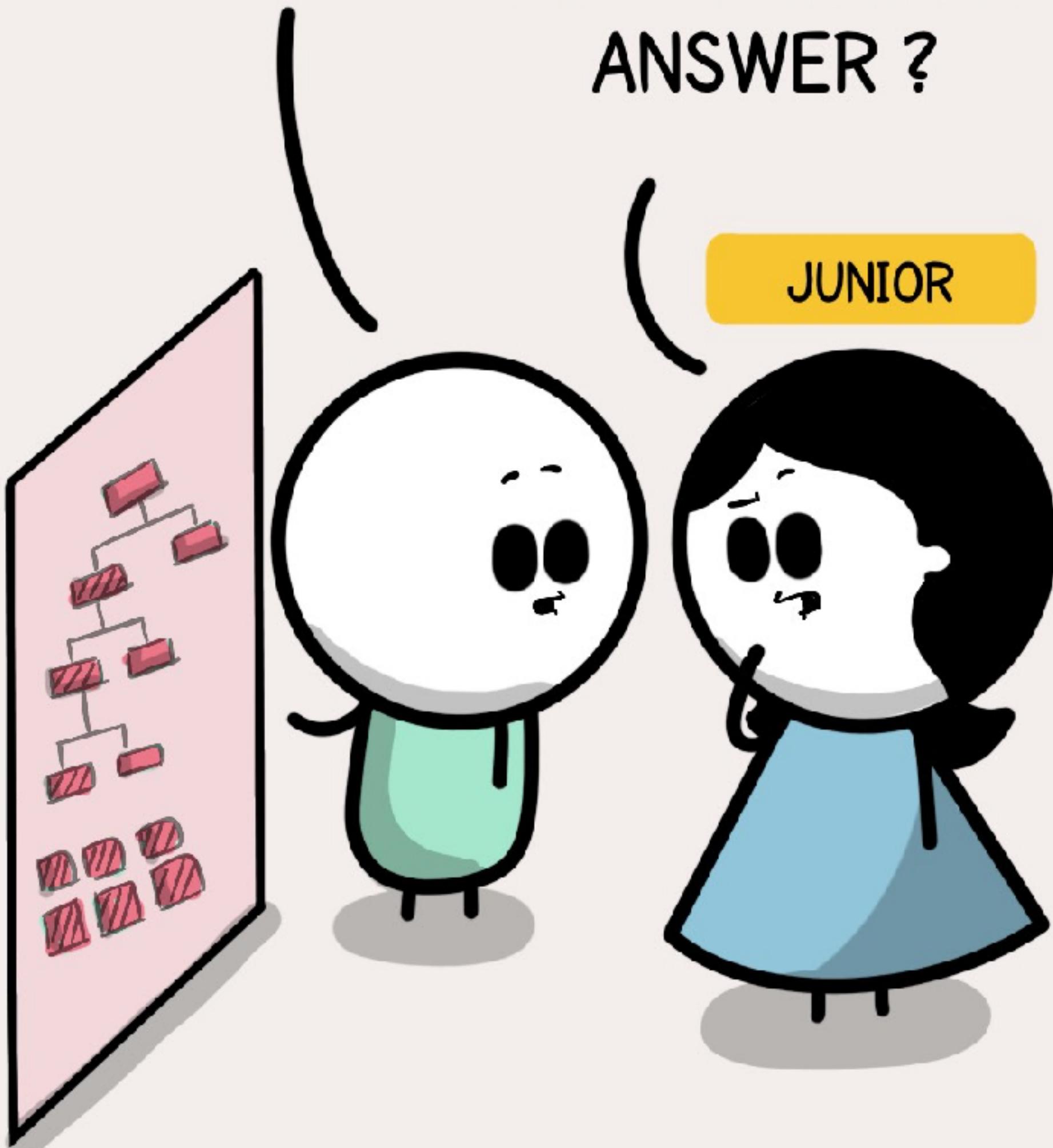
chaîne nombre décimal
 précision 2

protocol host port path optional data

```
http://foo.example:42/bar/baz?hello#plop
```

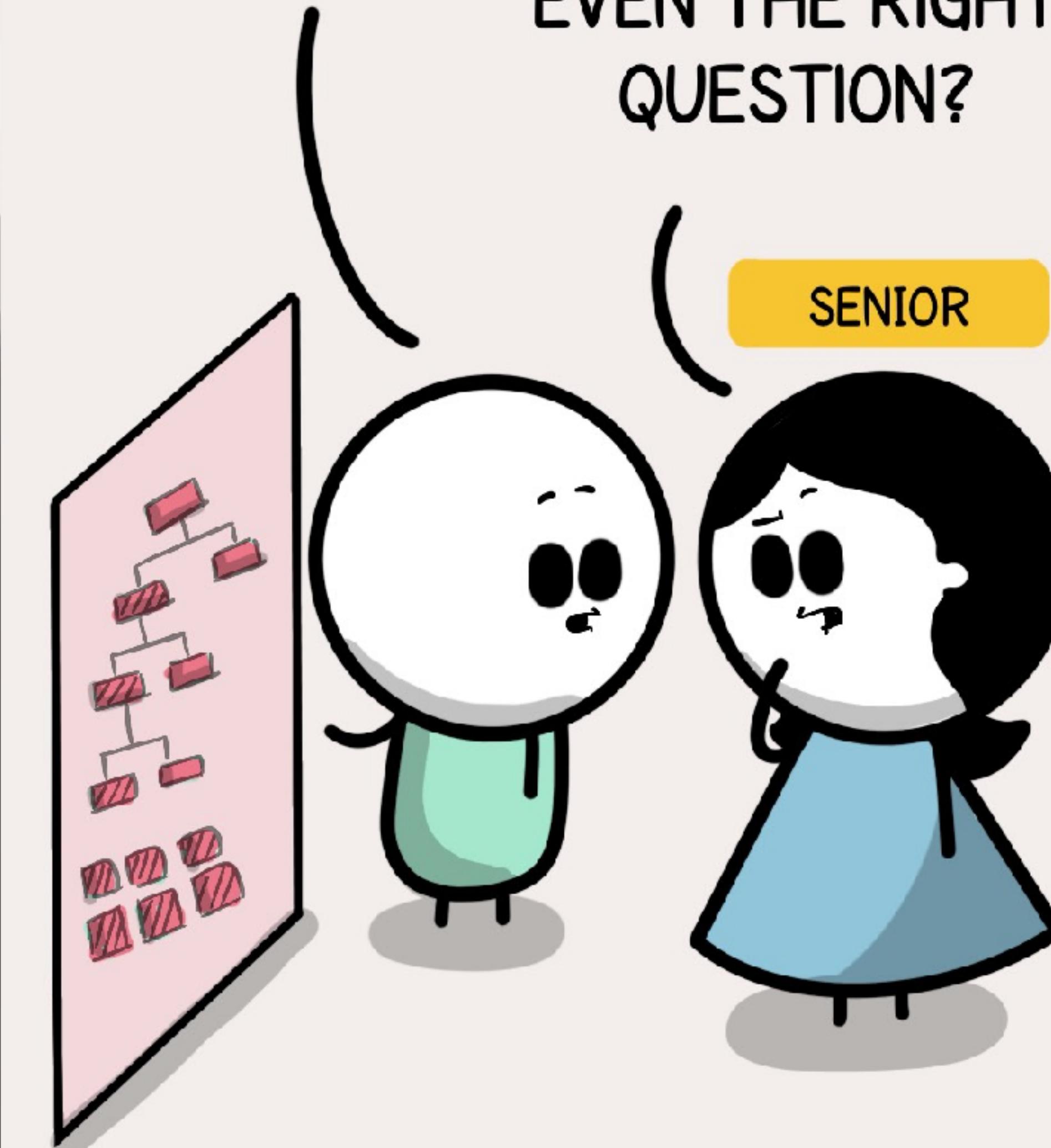
SO WHAT DO
YOU THINK?

HMM... IS
THIS THE RIGHT
ANSWER ?

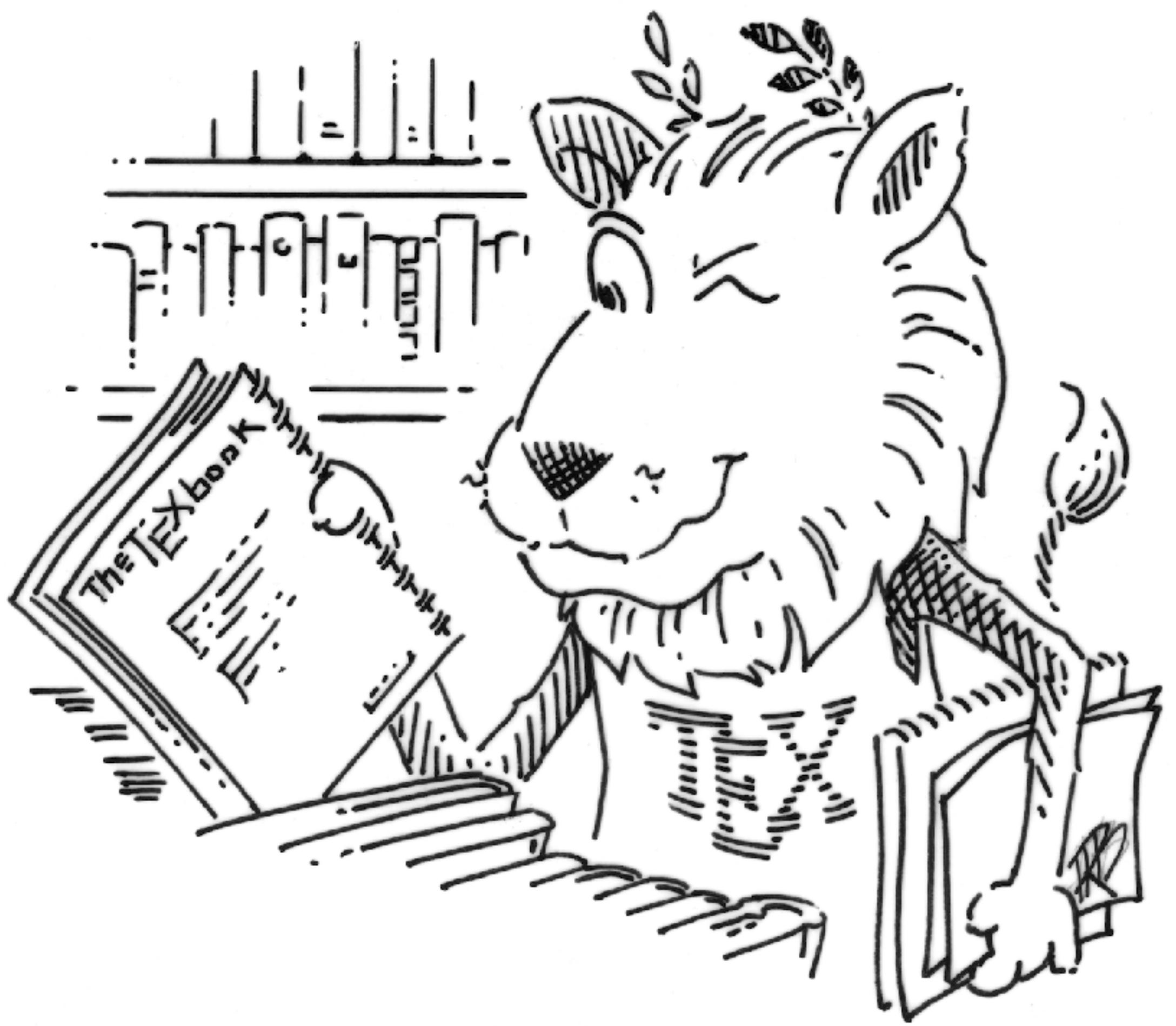


SO WHAT DO
YOU THINK?

HMM... IS THIS
EVEN THE RIGHT
QUESTION?



LATEX



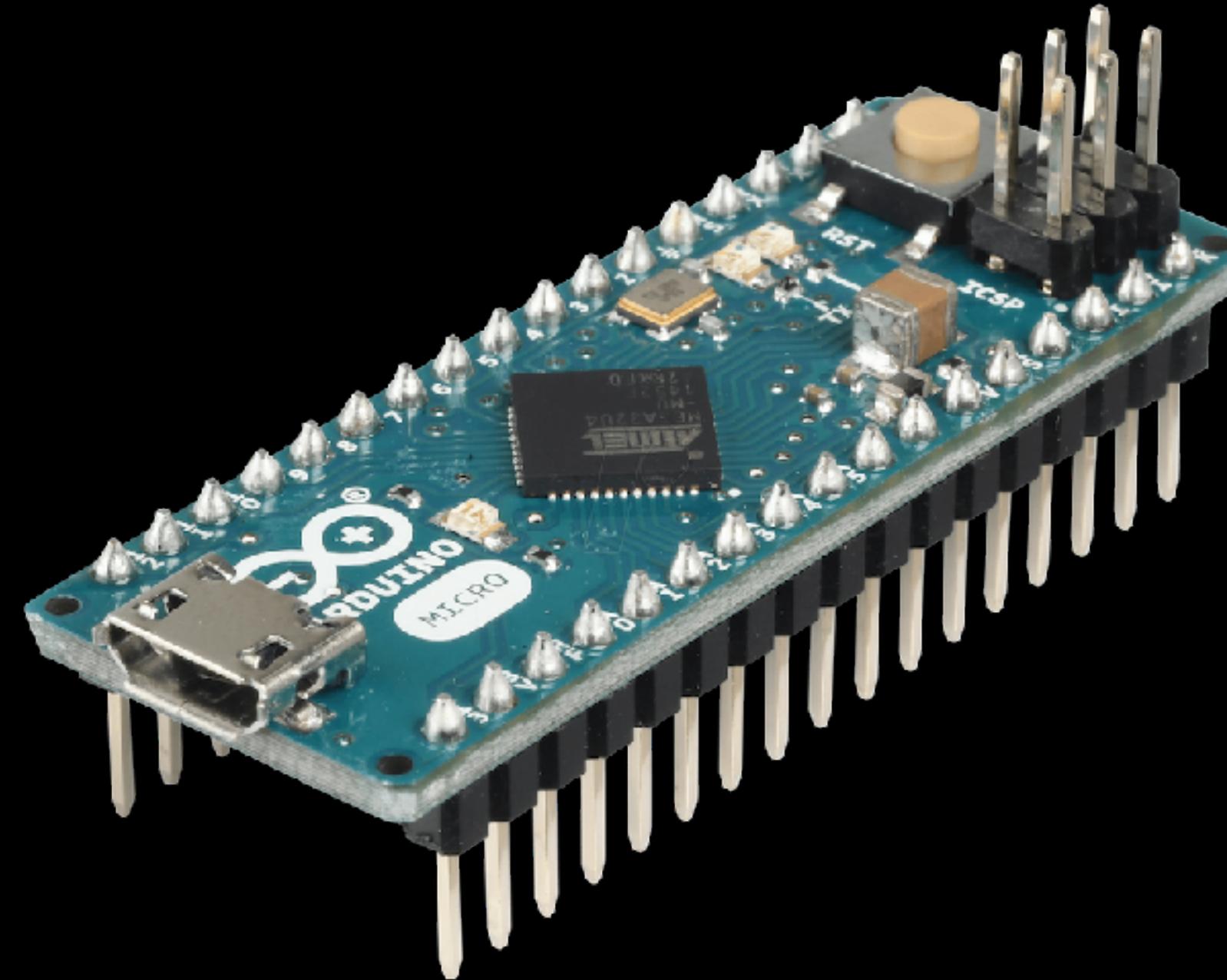
The avremu Package

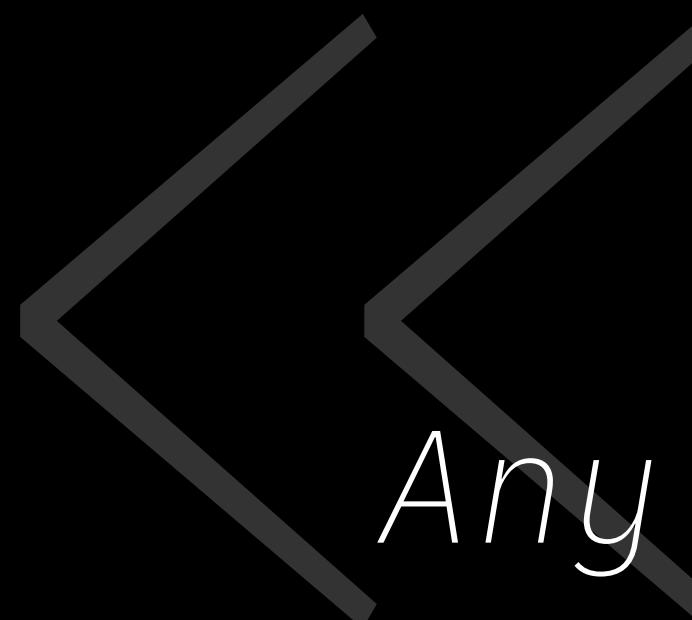
Christian Dietrich

stettberger@dokucode.de

<https://gitlab.brokenpipe.de/stettberger/avremu>

CTAN: <http://www.ctan.org/pkg/avremu>





Any sufficiently complicated C or Fortran program contains an ad hoc, informally-specified, bug-ridden, slow implementation of half of Common Lisp.

- Philip Greenspun's 10th rule (~1993)



*Almost every issue you'd encounter
in a full fledged general-purpose language
arises: types, inheritance, modularity,
FFI, functional vs. state*

- Gilad Bracha (about external DSLs)
DEX, Conflang 2021

Fluent Interface

return this;

+nommage

→ *construction par chaînage*

! fluent

```
void orderNormally(Customer customer) {  
    Order o = new Order();  
    customer.addOrder(o);  
    OrderLine l1 = new OrderLine(6, Product.find("TAL"));  
    o.addLine(l1);  
    OrderLine l2 = new OrderLine(5, Product.find("HPK"));  
    o.addLine(l2);  
    OrderLine l3 = new OrderLine(3, Product.find("LGV"));  
    o.addLine(l3);  
    l2.setSkippable(true);  
    o.setRush(true);  
}
```

fluent !

```
void orderFluently(Customer customer) {  
    customer.newOrder()  
        .with(6, "TAL")  
        .with(5, "HPK").skippable()  
        .with(3, "LGV")  
        .priorityRush();  
}
```

Expression Builder

```
class OrderBuilder {  
    Order subject = new Order();  
    OrderLine line;  
  
    Order getSubject() {  
        return subject;  
    }  
  
    OrderBuilder with(int qty,  
                      String prCode) {  
        line = new OrderLine(qty,  
                             Product.find(prCode));  
        subject.addLine(line);  
        return this;  
    }  
  
    OrderBuilder skippable() {  
        line.setSkippable(true);  
        return this;  
    }  
  
    OrderBuilder priorityRush() {  
        subject.setRush(true);  
        return this;  
    }  
}
```