

Intro

Talk:

title: Building your own DSL

date: 2017-09-21



community:

• GROOVY member (`Introlang.structure.COMMUNITY`)

• JAVA member (`Introlang.structure.COMMUNITY`)

Speaker:

name: Bohdan Levchenko ;

company: SoftServe Inc ;

email: blev@softserveinc.com ;

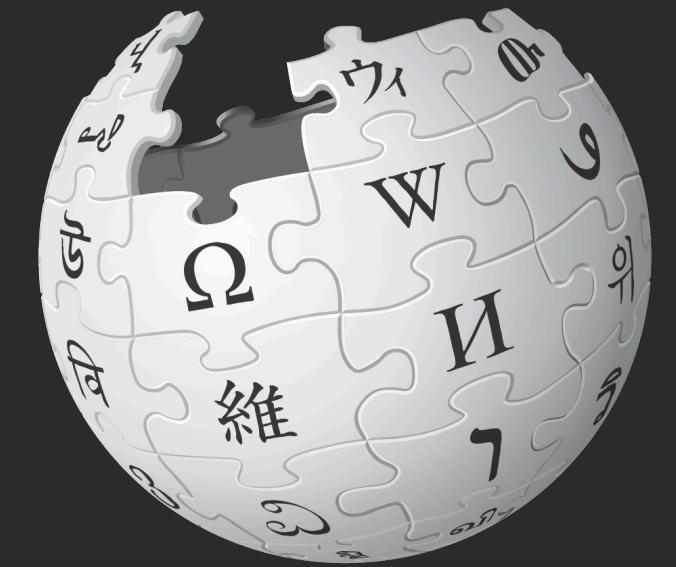
Plan for today

Agenda:

- * Fantastic DSLs and where to find them?
- * DSLs Kinds
- * Workbenches

What are DSL's?

A domain-specific language (DSL) is a computer language specialized to a particular application domain. This is in contrast to a general-purpose language (GPL), which is broadly applicable across domains. There is a wide variety of DSLs, ranging from widely used languages for common domains down to languages used by only one or a few pieces of software.



What are DSL's, again?

Specialized mini languages for specific problem domain
that make it easier to work in that particular domain.

What are DSL's, again?

Specialized mini languages for specific problem domain that make it easier to work in that particular domain.

Properties

- * Syntax is extremely focused on domain
- * Very limited superset of capabilities
- * Tend to be plain and compact

Well known example - SQL

```
SELECT  
    e.dept_id,  
    count(e.id)  
FROM employees e  
WHERE e.dept_id IN (204810, 192010, 160913)  
GROUP BY e.dept_id  
HAVING count(e.id) > 0|
```

Well known example - CSS

```
.feed-title {  
    font-weight: bold;  
    text-align: center;  
    color: #cafebabe;  
    padding: 10px;  
}
```

Well known example - Gradle

```
buildscript {  
    ext {  
        springBootVersion = '1.4.2.RELEASE'  
    }  
    repositories {  
        mavenCentral()  
    }  
}  
  
apply plugin: 'java'  
sourceCompatibility = 1.8  
targetCompatibility = 1.8  
  
dependencies {  
    compile('org.springframework.boot:spring-boot-starter-actuator')  
    compile('org.springframework.boot:spring-boot-starter-web')  
    compile('org.springframework.boot:spring-boot-starter-data-jpa')  
    compile('org.springframework.boot:spring-boot-starter-cache')  
    compile('org.springframework.boot:spring-boot-starter-validation')  
    compile('org.springframework.boot:spring-boot-starter-mail')  
    compile('org.springframework.boot:spring-boot-starter-security')  
}
```

Makefile (o_O)?

```
CC=cc -I.  
▶ all: clean compile  
▶     compile: demo  
▶         demo: lib.o  
▶             $(CC) -o demo demo.cc lib.o  
▶ lib.o:  
▶     $(CC) -c lib.cc lib.h  
▶ clean:  
▶     /bin/rm -f demo lib.o
```

Why DSL?

When working with RDBMS – use SQL

When working with styles – use CSS

When working with build configuration – use Makefile/gradle/ant

Why DSL?

When working with RDBMS – use SQL

When working with styles – use CSS

When working with build configuration – use Makefile/gradle/ant

When working with your domain – use your own DSL!

Domain... domain... DDD?

****Context****

The setting in which a word or statement appears that determines its meaning;

****Domain****

A sphere of knowledge (ontology), influence, or activity.

The subject area to which the user applies a program is the domain of the software;

****Model****

A system of abstractions that describes selected aspects of a domain and can be used to solve problems related to that domain;

****Ubiquitous Language****

A language structured around the domain model and used by all team members to connect all the activities of the team with the software.

Domain... domain... DDD?



DSLs Kinds



DSLs Kinds: External vs Internal

External

- + You have full control on your lang, unlimited flexibility
- + Challenging to write!
- You have to write a toolkit (parser/lexer) for your lang
- Easy to grow out of control

Internal

- + Using already familiar lang
- + No need to learn how lang toolkit works or write your own
- + Native interoperability with host lang
- Your lang will inherit all the limitation of the host lang

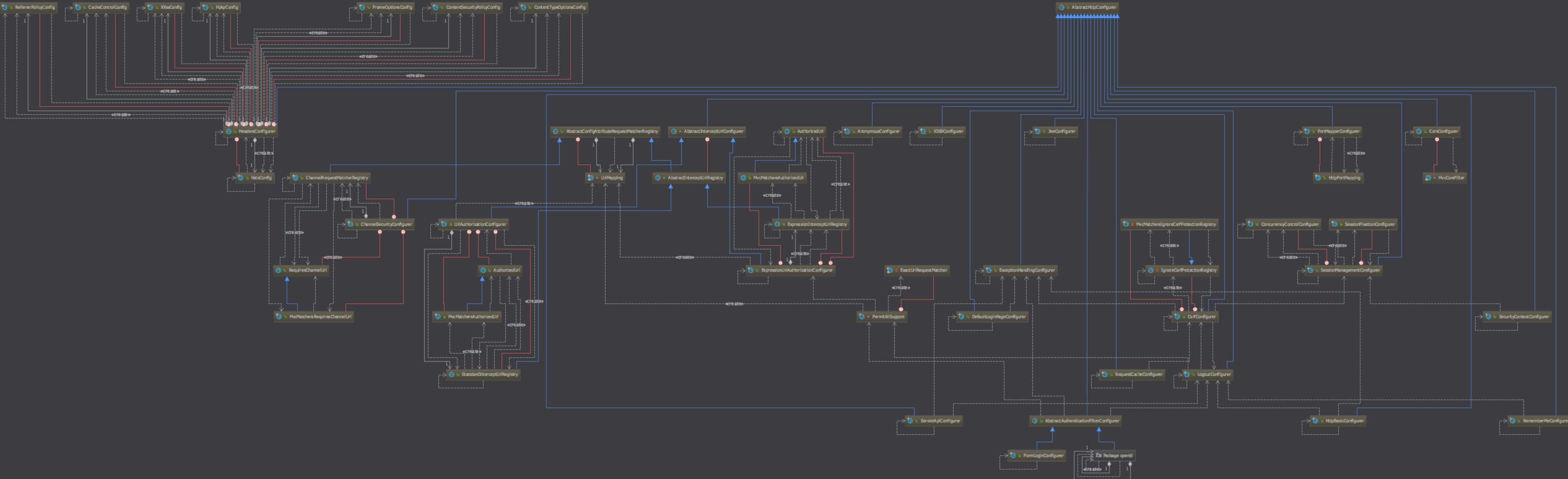
Groovy DSL

HOST LANGUAGE?

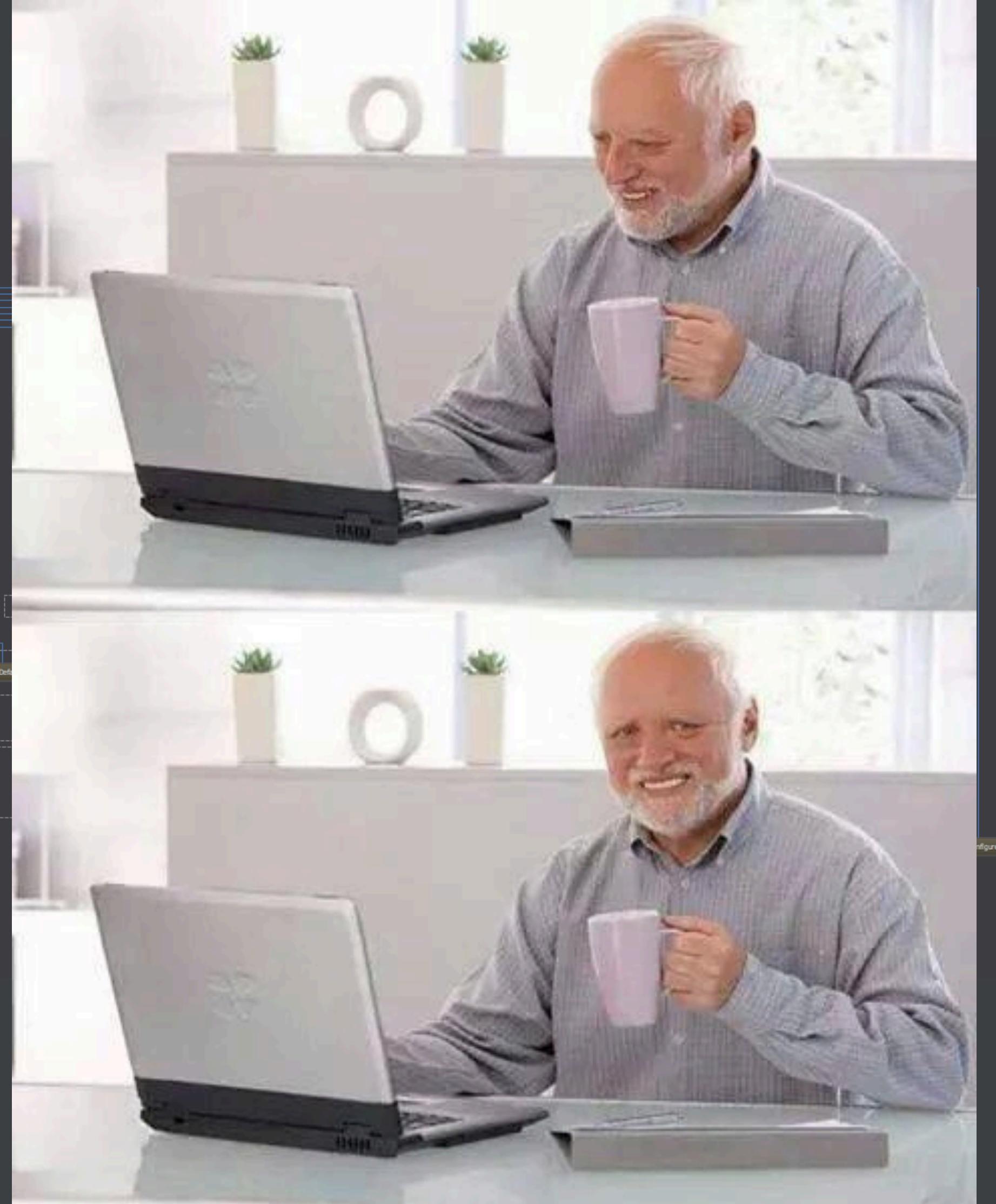
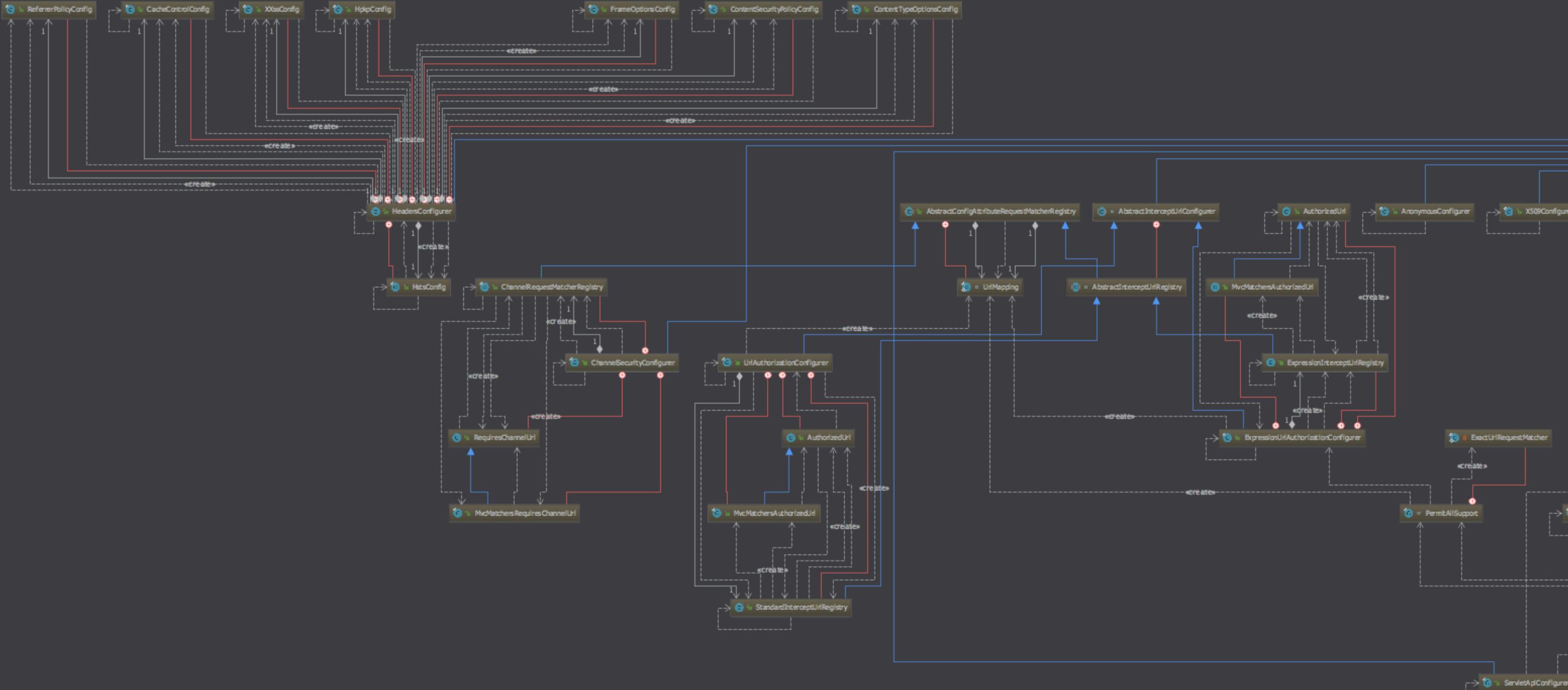
Internal DSL on Java

```
@Override  
protected void configure(HttpSecurity http) throws Exception {  
    http  
        .csrf().disable()  
        .httpBasic().realmName("MYAPP")  
        .and()  
        .formLogin()  
            .loginPage("/login")  
            .usernameParameter("email")  
            .passwordParameter("password")  
            .failureForwardUrl("/login?error=1&email=%s")  
        .and()  
        .logout()  
            .logoutUrl("/logout")  
            .clearAuthentication(true)  
        .and()  
        .authorizeRequests()  
            .mvcMatchers("/public/**").permitAll()  
            .mvcMatchers("/protected/**").fullyAuthenticated()  
            .mvcMatchers("/protected/admin/**").access("hasRole('ROLE_ADMIN')")  
            .mvcMatchers("/protected/supervisor/**").access("hasRole('ROLE_SUPERVISOR')")  
        .and()  
        .sessionManagement()  
            .maximumSessions(1)  
            .expiredUrl("/expired")  
        .and()  
        .sessionCreationPolicy(SessionCreationPolicy.STATELESS);  
}
```

Internal DSL on Java (under the hood)



Internal DSL on Java (under the hood)



Groovy DSL



Live coding

DEMO TIME

Q&A?

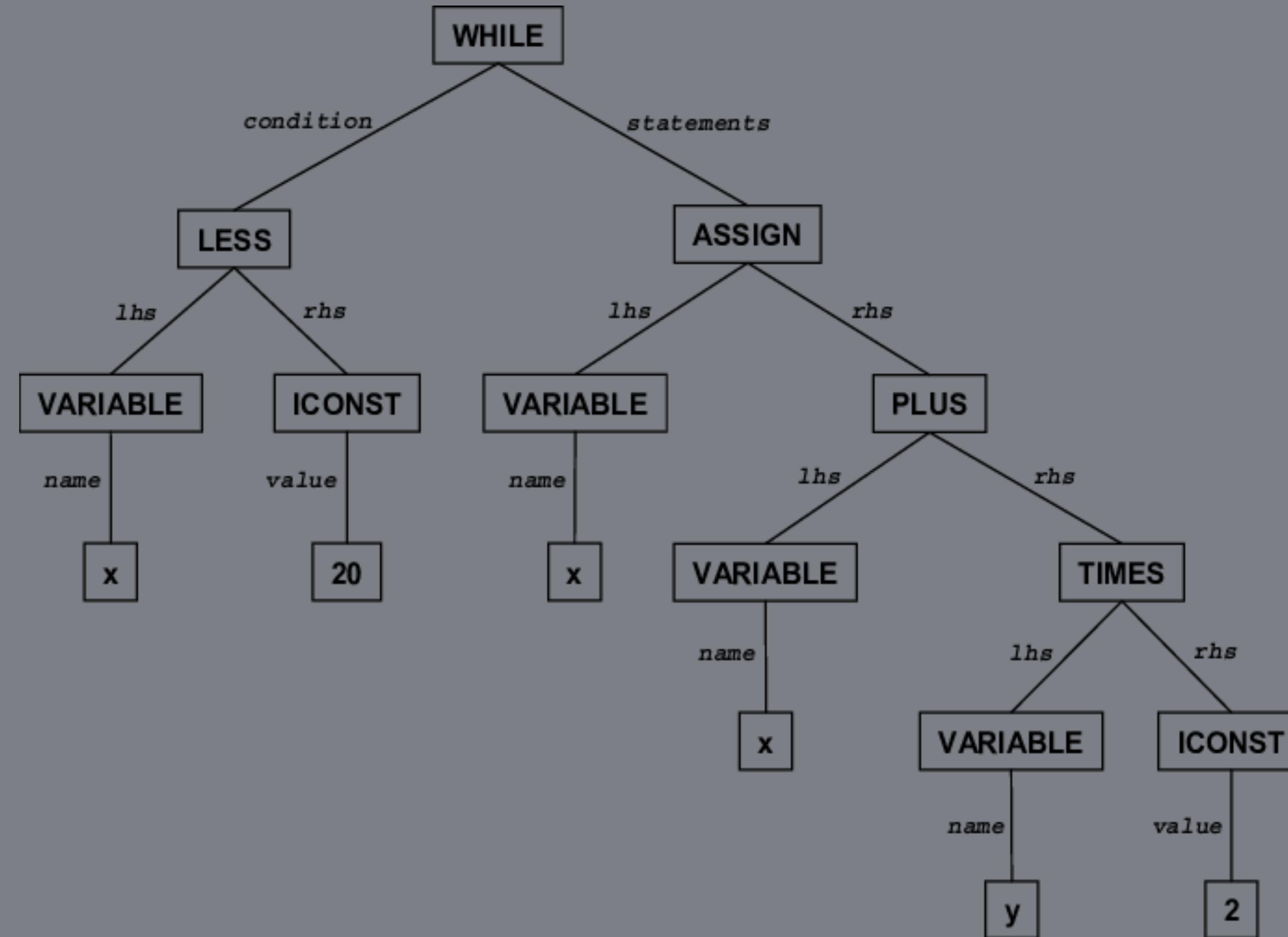


Language workbenches



JetBrains MPS

AST - Abstract Syntax Tree



Live coding

DEMO TIME

Math

```
public class Sample {  
  
    public static void main(String[] args) {  
        double a = Math.random();  
        double b = Math.random();  
        List<Integer> list = new ArrayList<Integer>{1, 2, 3, 4};  
  
        k = 0  
        System.out.println(String.valueOf((sum(99, matrix<Double> s =  
            [3.0, sin(1), 1, 1,  
             2, 1, 3 + 1.0 / 2, 2,  
             3, 7 - 1.0 / 2 + 1, exp(1), 3,  
             0, 2, 0, 0;  
            [4, 0, 0, 0],  
            [cos(a), sin(1), 1,  
             2, 1, 3 + 1.0 / 2,  
             3, 7 - a / 2 + b, exp(1),  
             0, 0, prod(n)],  
            MAX 3 * ln(e),  
            e in list  
            [4, 0, 0, 0],  
            [cos(a), sin(1), 1,  
             2, 1, 3 + 1.0 / 2,  
             3, 7 - a / 2 + b, exp(1),  
             0, 0, prod(n)]  
        ));  
        System.out.println(exp(a + i * b) - exp(a) * (cos(b) + i * sin(b)));  
        System.out.println(String.valueOf((sum(99, matrix<Double> r =  
            [3.0, sin(1), 1, 1,  
             2, 1, 3 + 1.0 / 2, 2,  
             3, 7 - a / 2 + b, exp(1), 3,  
             0, 2, 0, 0;  
            [4, 0, 0, 0],  
            [cos(a), sin(1), 1,  
             2, 1, 3 + 1.0 / 2,  
             3, 7 - a / 2 + b, exp(1),  
             0, 0, prod(n)],  
            MAX 3 * ln(e),  
            e in list  
            [4, 0, 0, 0],  
            [cos(a), sin(1), 1,  
             2, 1, 3 + 1.0 / 2,  
             3, 7 - a / 2 + b, exp(1),  
             0, 0, prod(n)]  
        ));  
        System.out.println(s);  
        System.out.println(r);  
    }  
}
```

Smart home

Heating plan for room: Kitchen

Daily Plan applicable from Monday to Sunday

At 6 o'clock set temperature to 23 Celsius

At 10 o'clock set temperature to 22 Celsius

At 19 o'clock set temperature to 20 Celsius

Daily Plan applicable on weekend day customizing the From monday to sunday plan

customizing At 10 o'clock set temperature to 21 Celsius

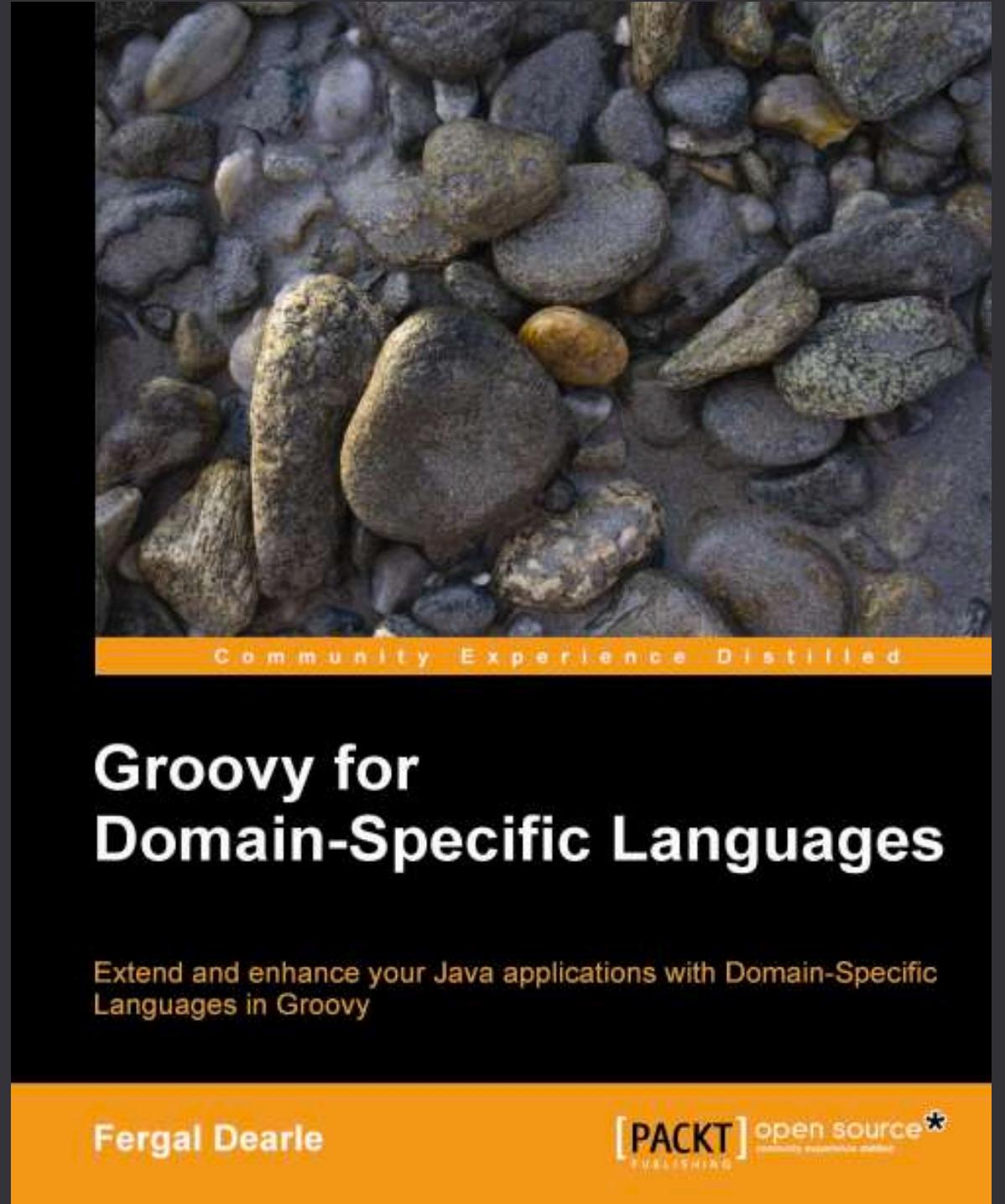
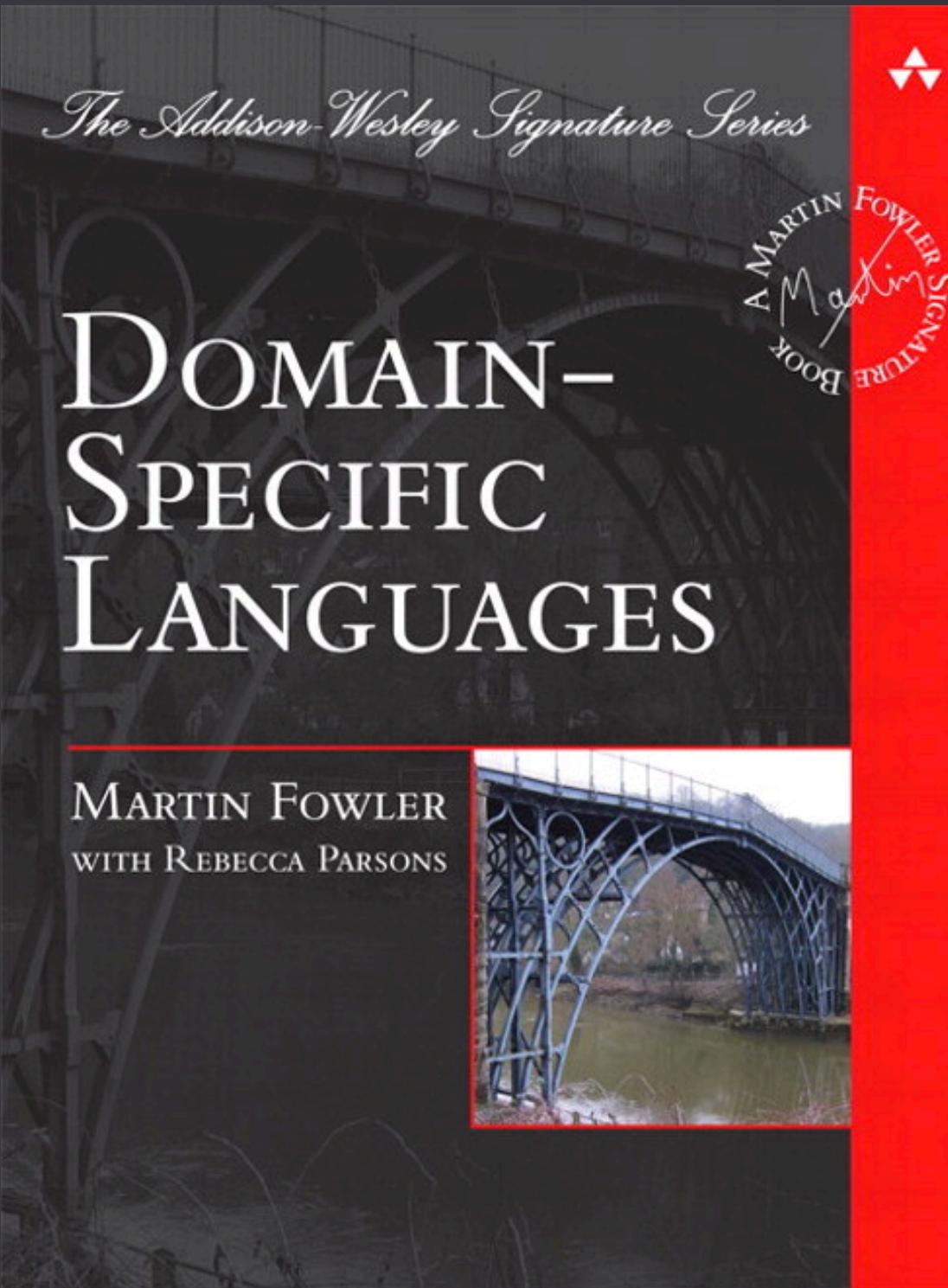
At 12 o'clock set temperature to 22 Celsius

Money

```
public final class Sample {  
    public static void main(String[] args) {  
        Money m1 = 10 USD;  
        Money m2 = 20 EUR;  
        System.out.println("Result: " + (m2 - m1));  
    }  
}
```



Readings



- Build your own DSL

- @bohdanlevchenko

{Pacemaker 2017}

That's all folks!

