NODEWORLD

ADVENTURES FROM A WORLD OF TURTLES ALL THE WAY DOWN

Markus Voelter
voelter@acm.org

All-Turtle-Systems in History

Unix Shell
- All Text Files, Processes and Pipes

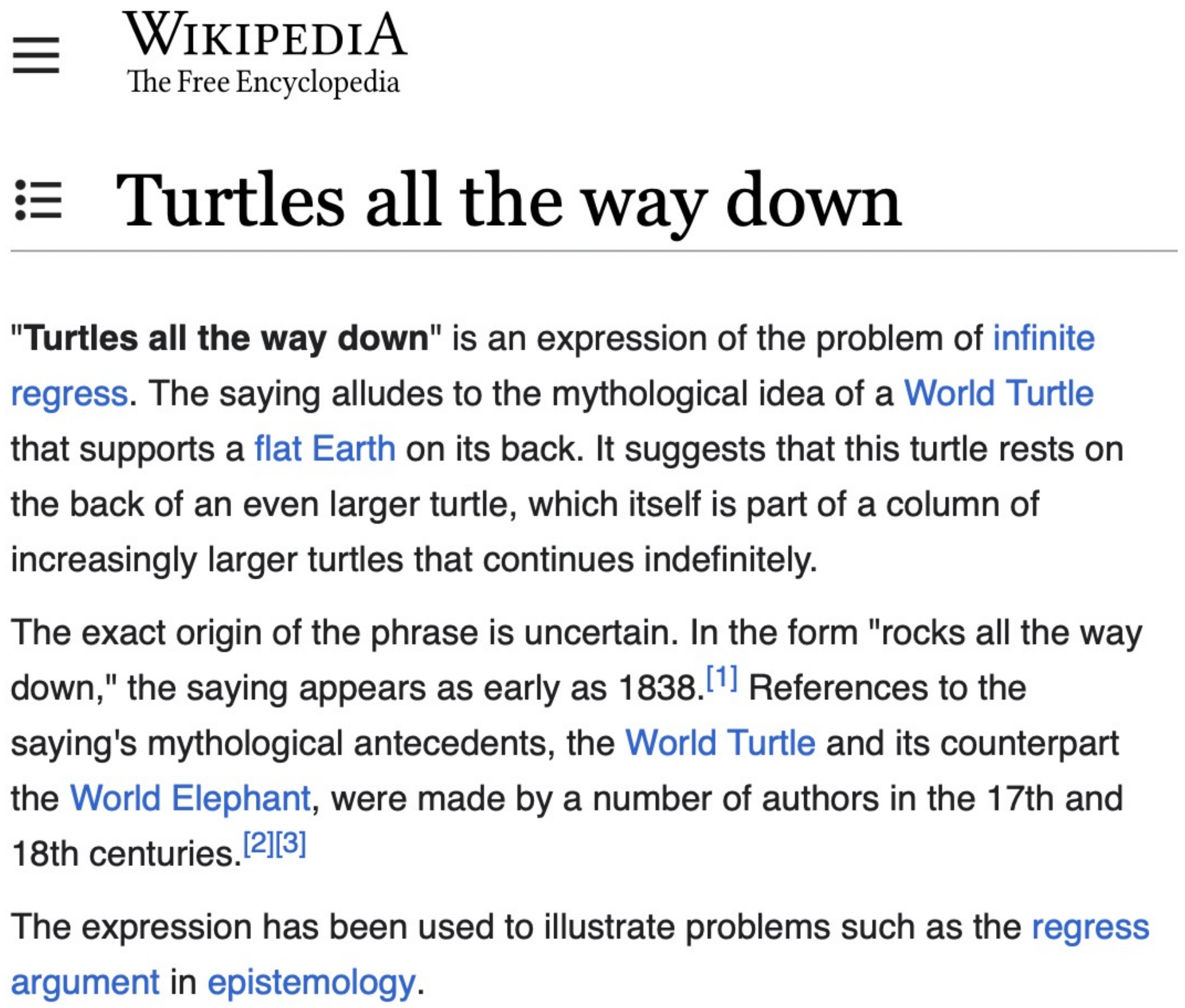
Smalltalk
- Everything is objects, including code

Lisp
- Everything is values, lists, functions

MPS
- Everything is nodes

BECAUSE EVERYTHING IS THE SAME, YOU NEED ONLY FEW GENERIC TOOLS TO DO "EVERYTHING"

Where does TATWD come from?



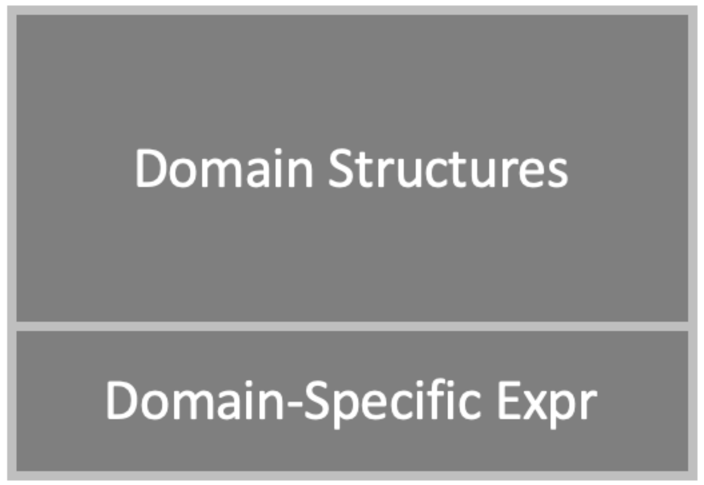
Wikipedia
The Free Encyclopedia

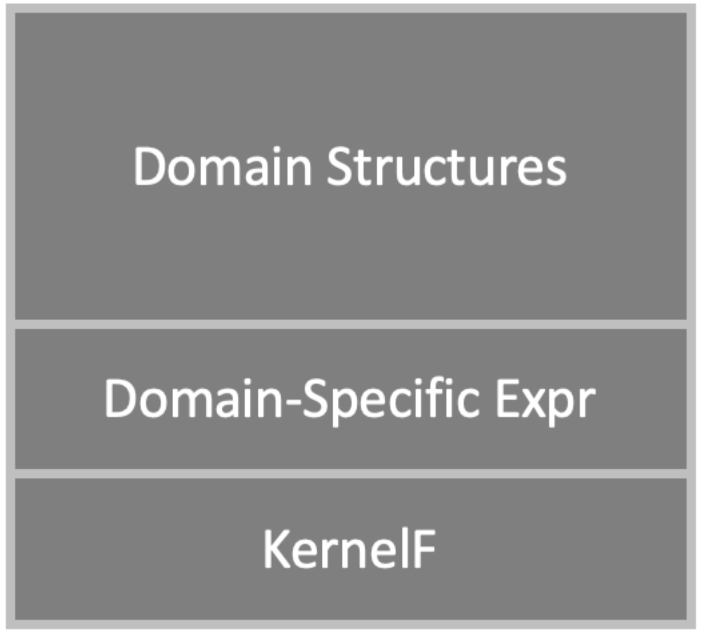
Turtles all the way down

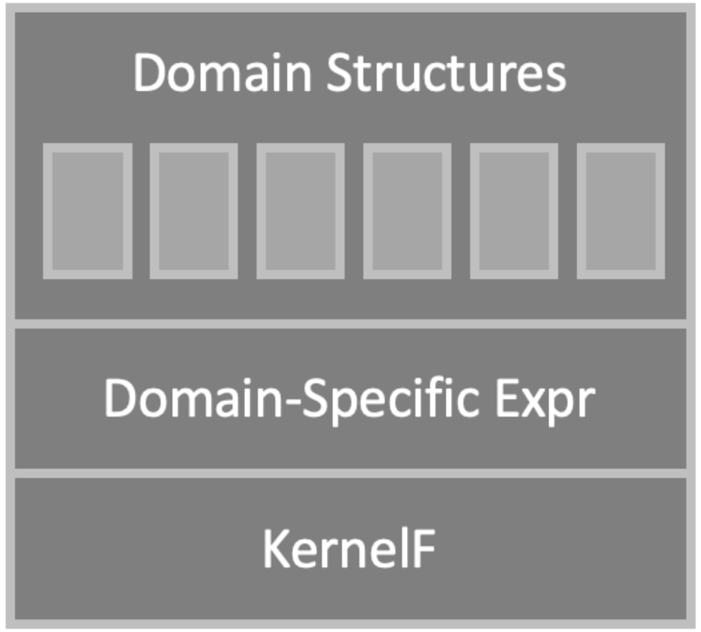
"**Turtles all the way down**" is an expression of the problem of infinite regress. The saying alludes to the mythological idea of a World Turtle that supports a flat Earth on its back. It suggests that this turtle rests on the back of an even larger turtle, which itself is part of a column of increasingly larger turtles that continues indefinitely.

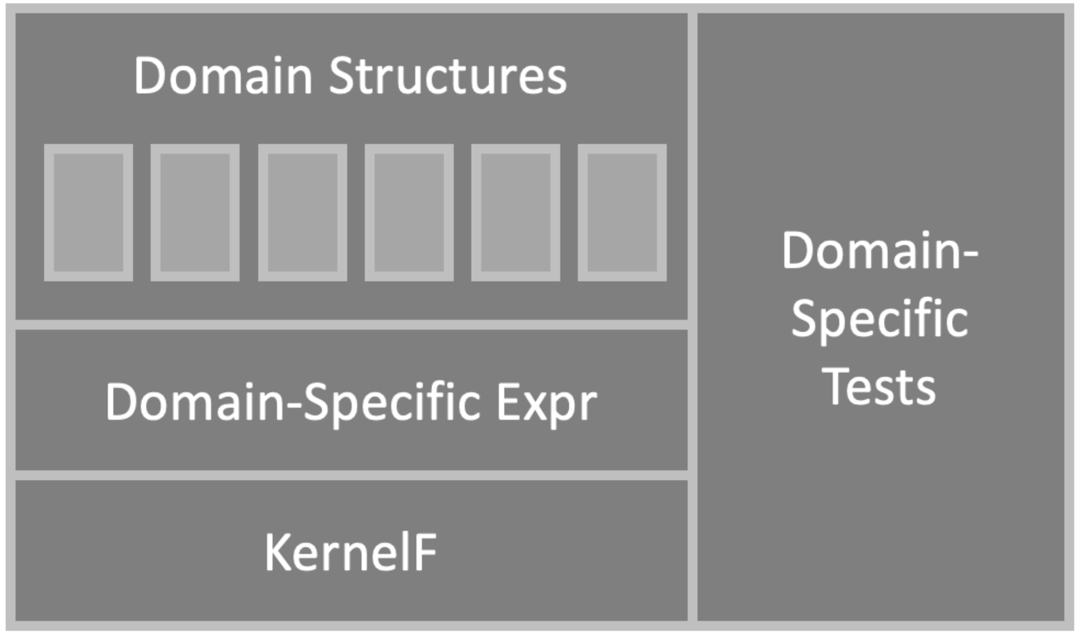
The exact origin of the phrase is uncertain. In the form "rocks all the way down," the saying appears as early as 1838.[1] References to the saying's mythological antecedents, the World Turtle and its counterpart the World Elephant, were made by a number of authors in the 17th and 18th centuries.[2][3]

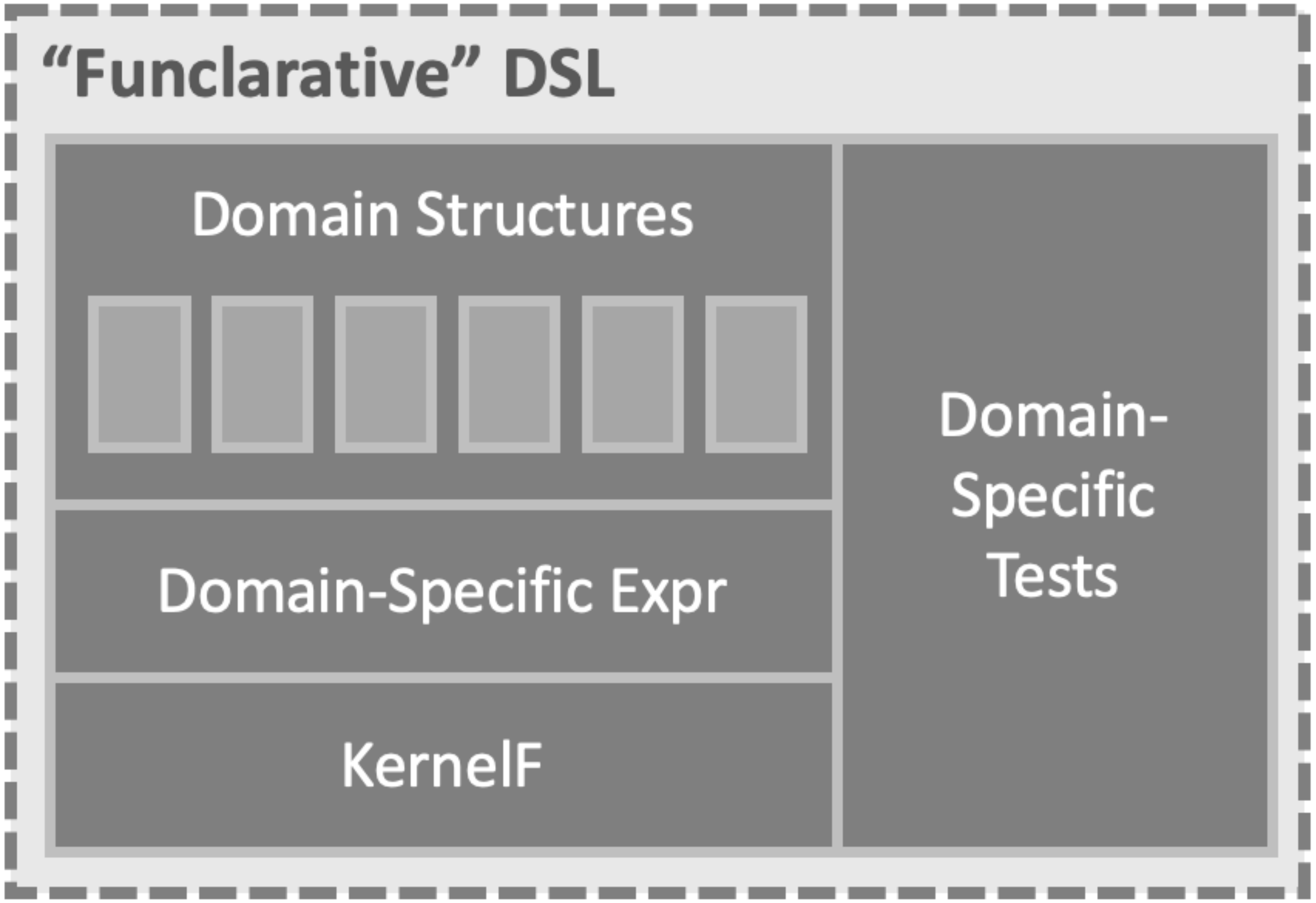
The expression has been used to illustrate problems such as the regress argument in epistemology.

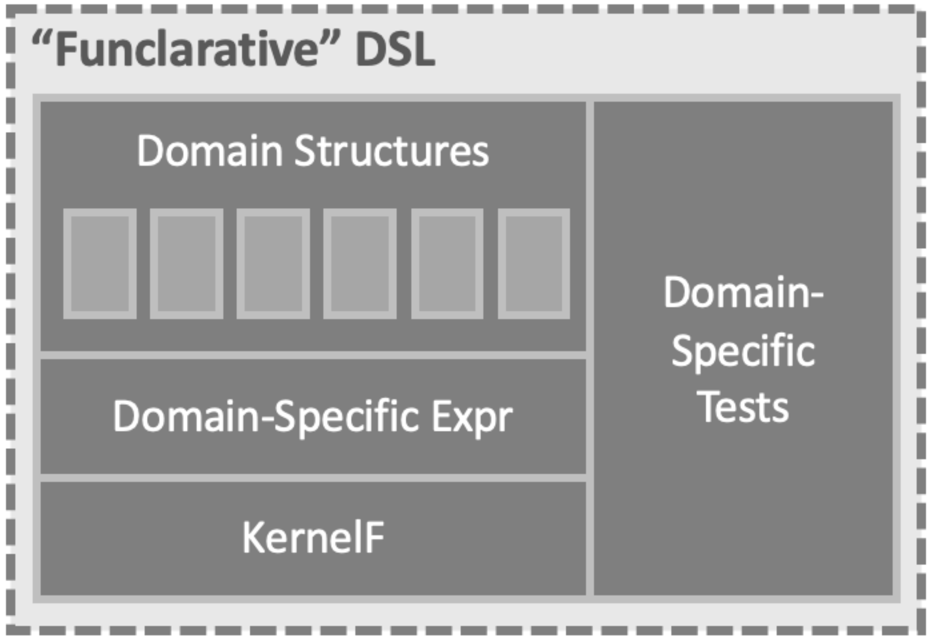

Domain Structures
Domain-Specific Expr

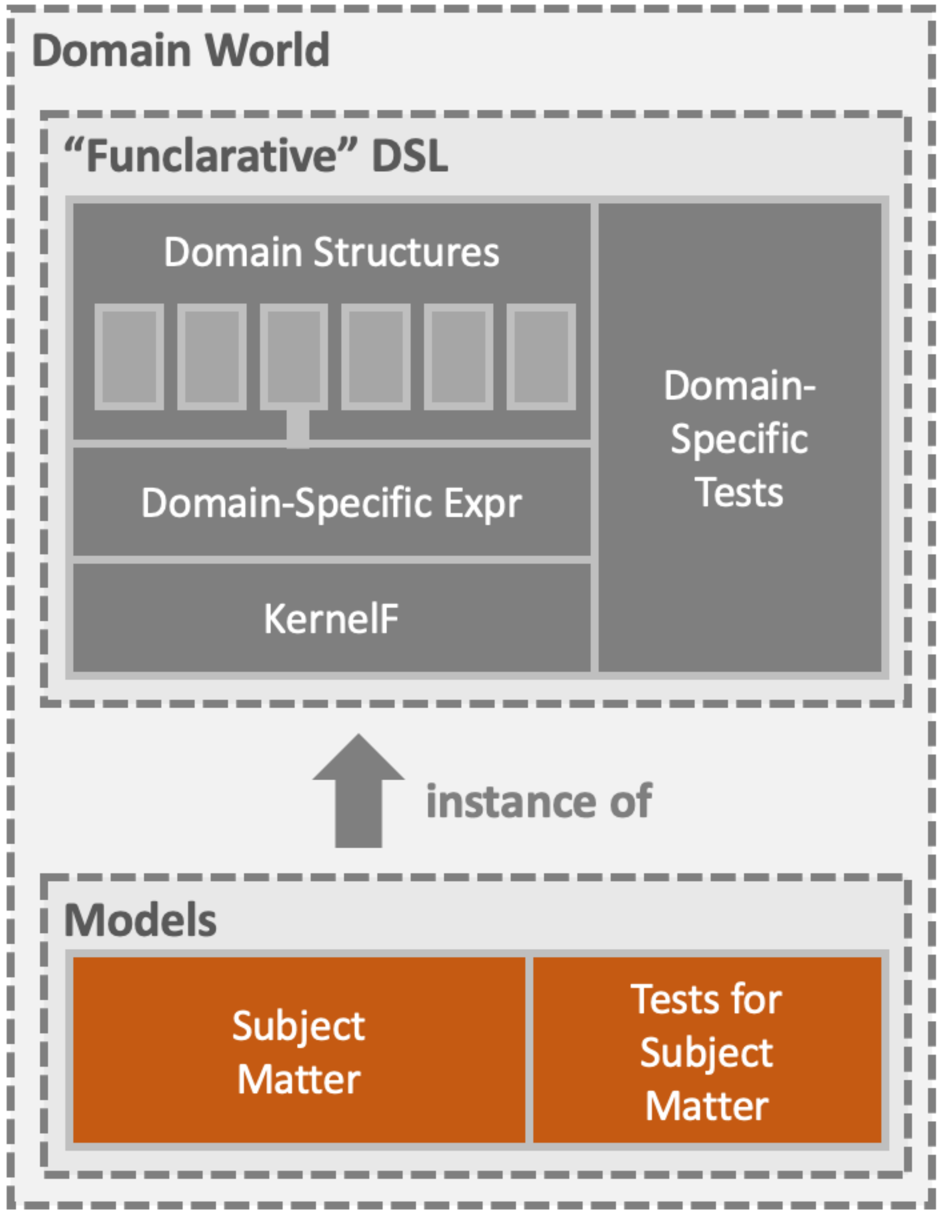

Domain Structures
Domain-Specific Expr
KernelF

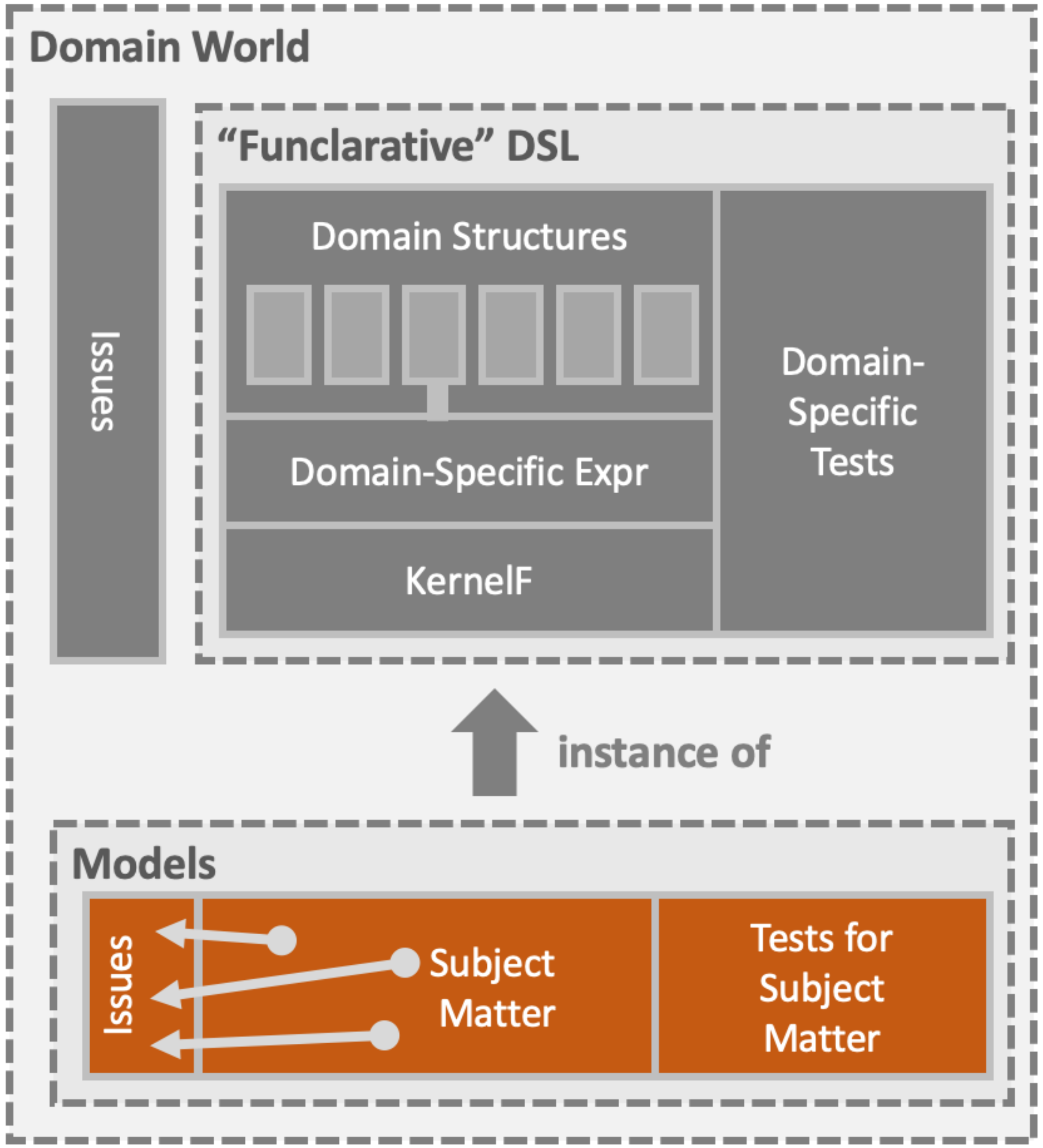

Domain Structures
Domain-Specific Expr
KernelF

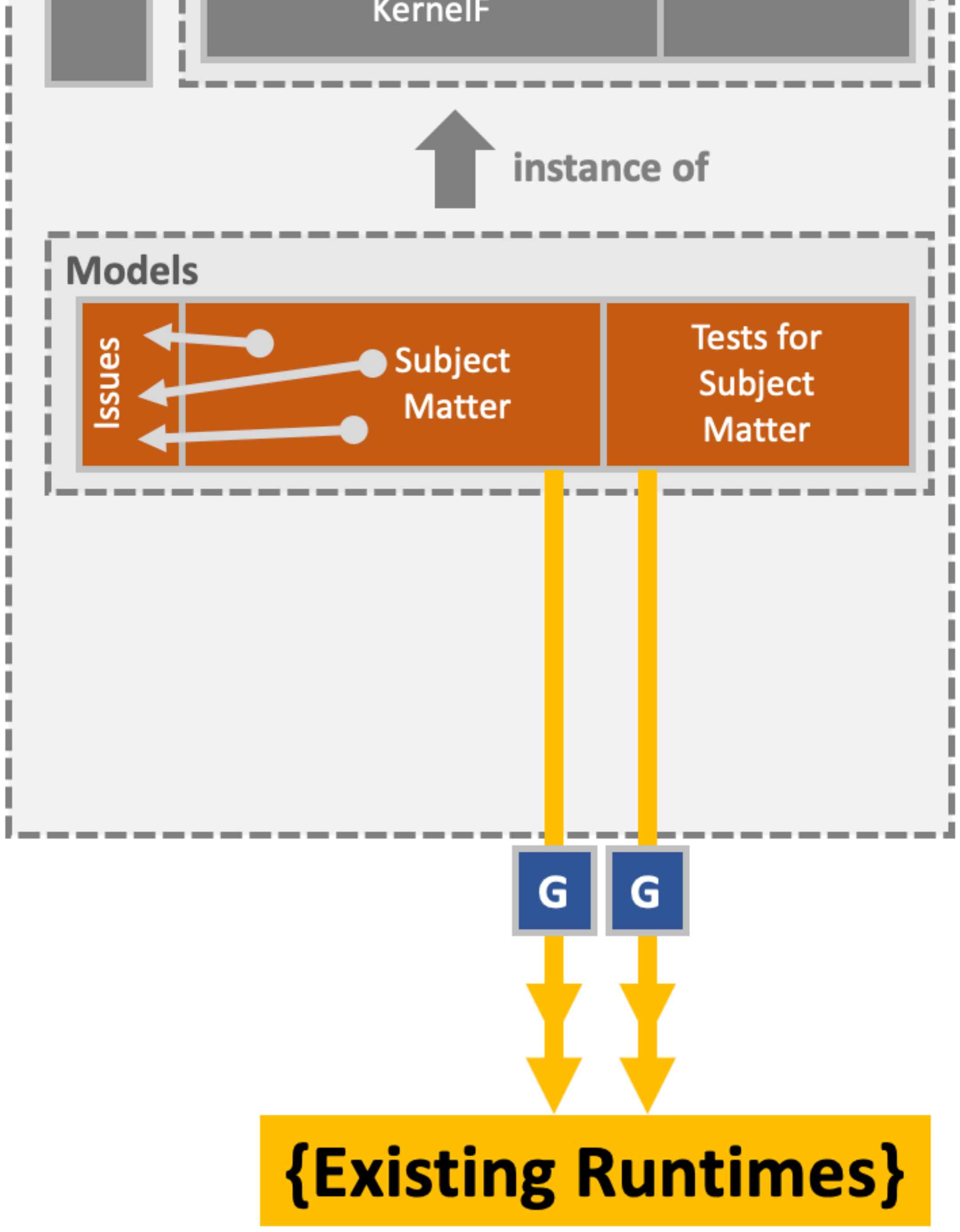

Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests

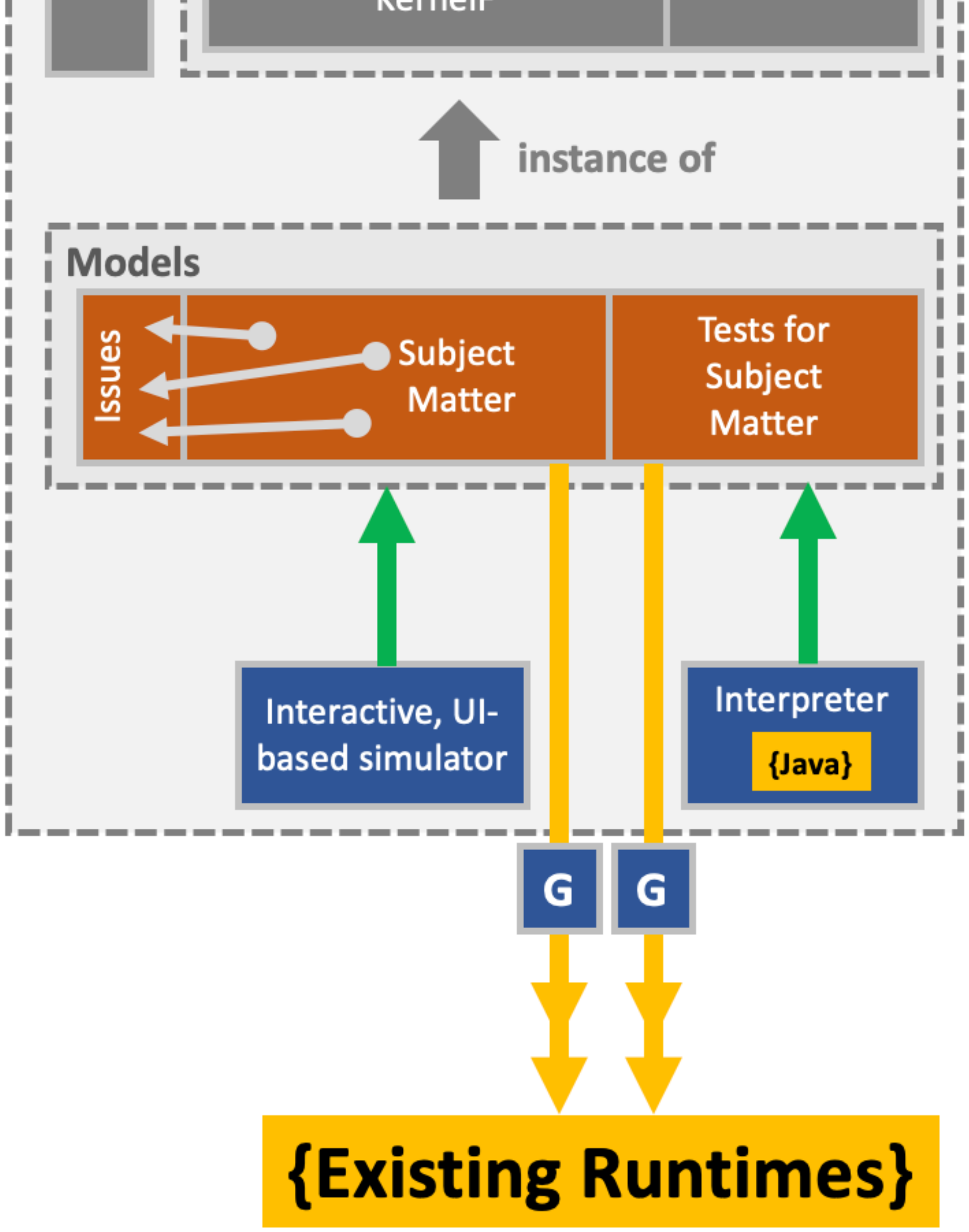

"Funclarative" DSL
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests

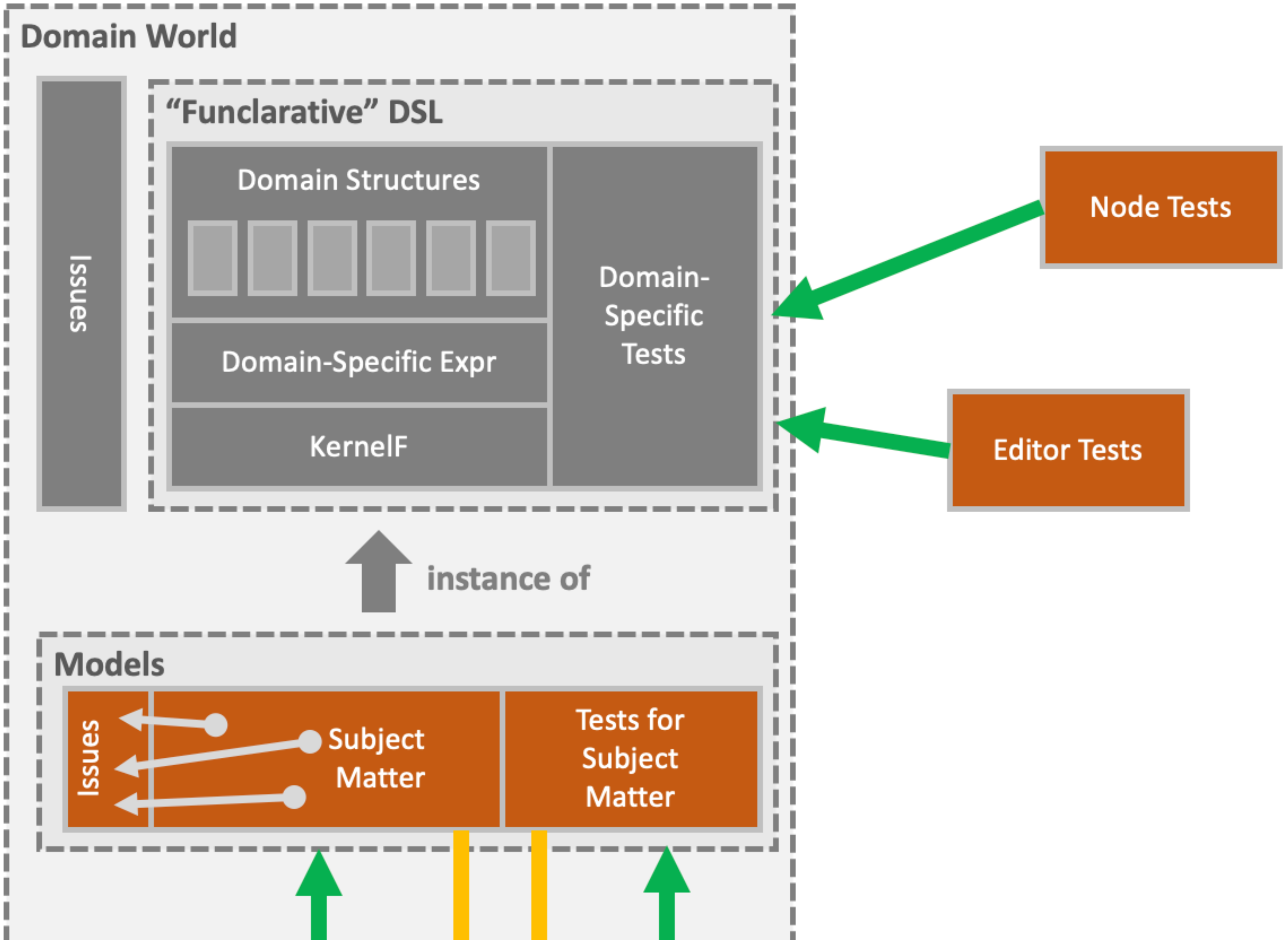

"Funclarative" DSL
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests

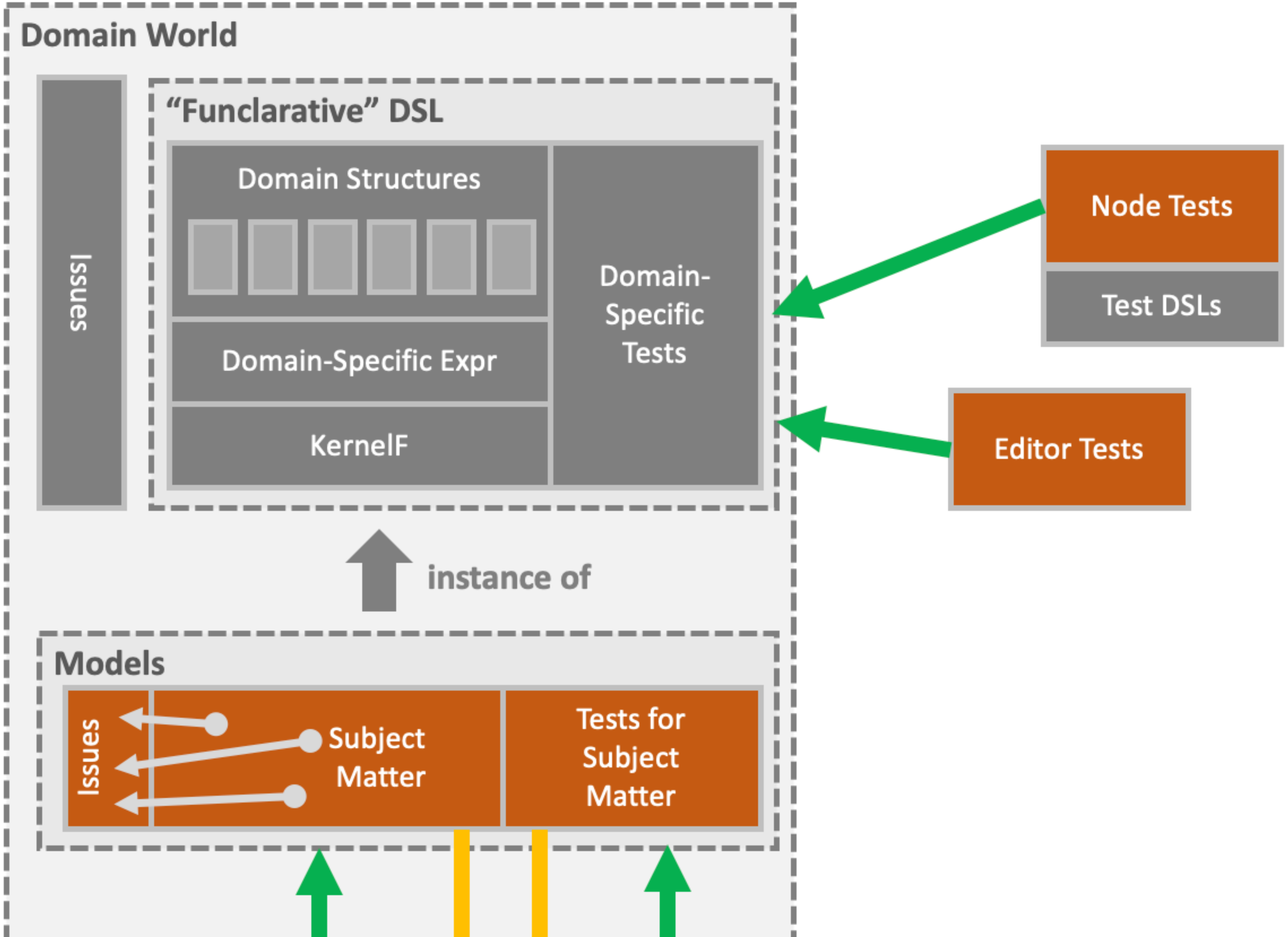

Domain World
"Funclarative" DSL
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
instance of
Models
Subject Matter
Tests for Subject Matter

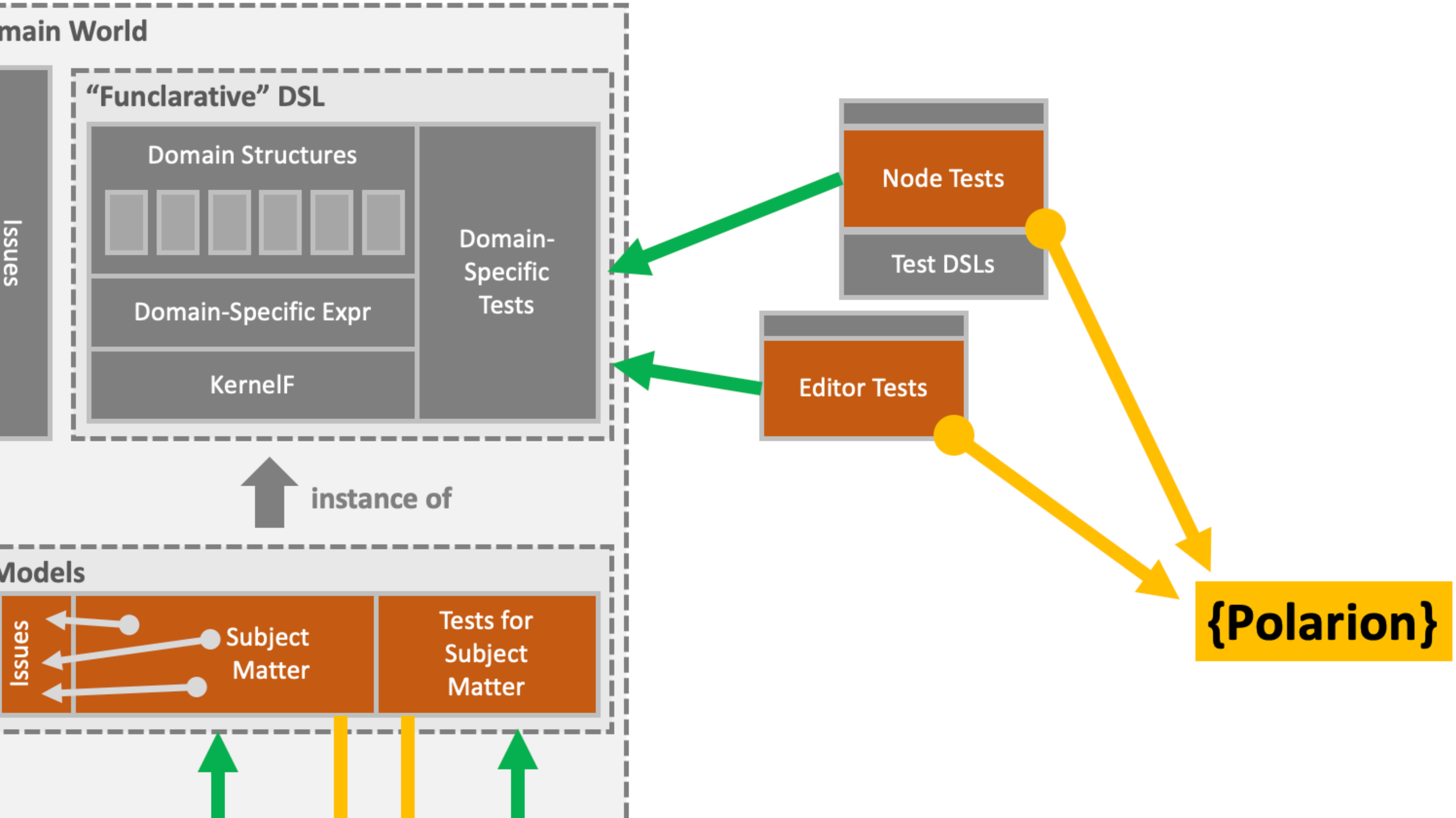

Domain World
Issues
"Funclarative" DSL
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
instance of
Models
Issues
Subject Matter
Tests for Subject Matter

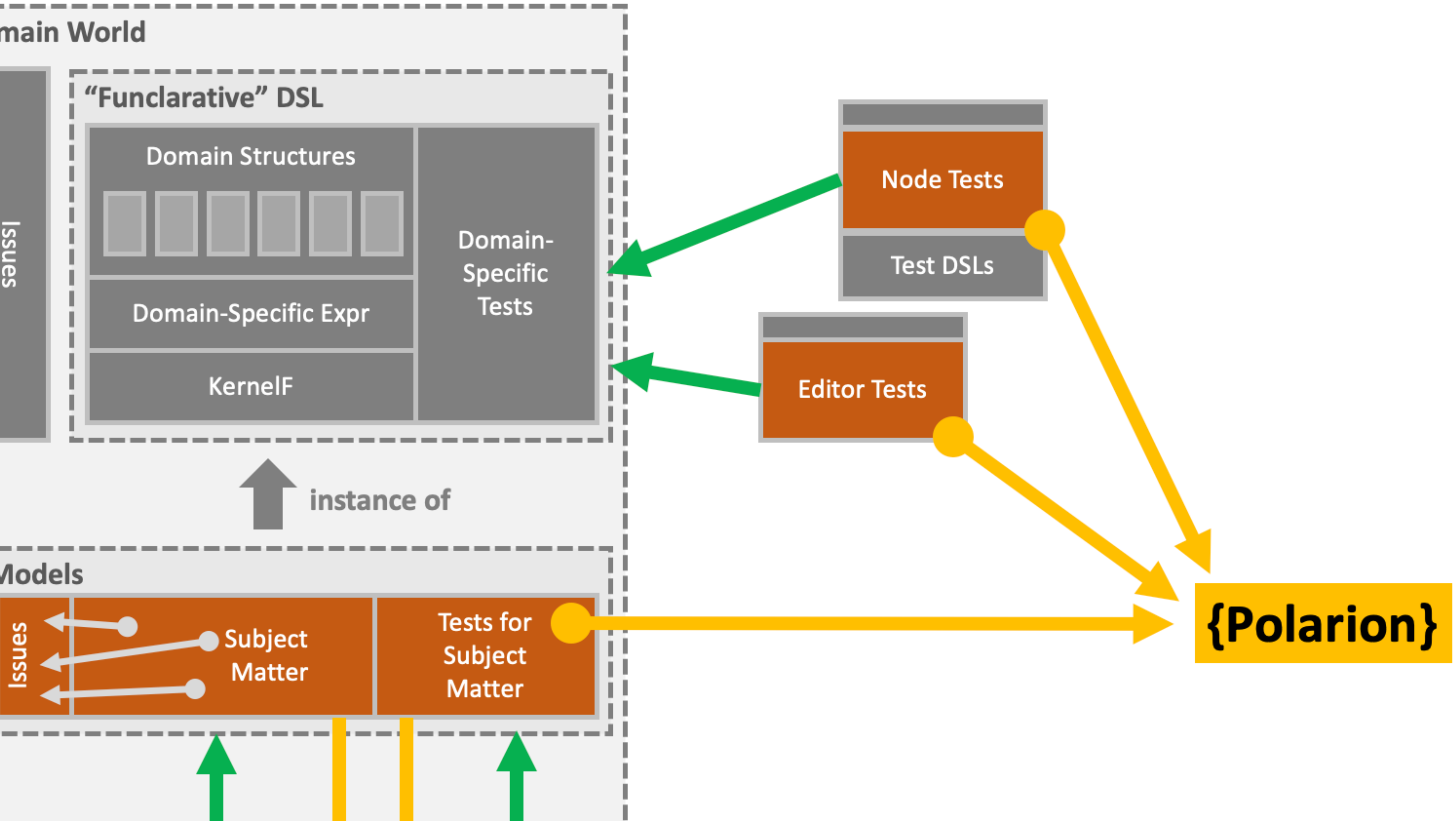

KernelF
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
G
G
{Existing Runtimes}

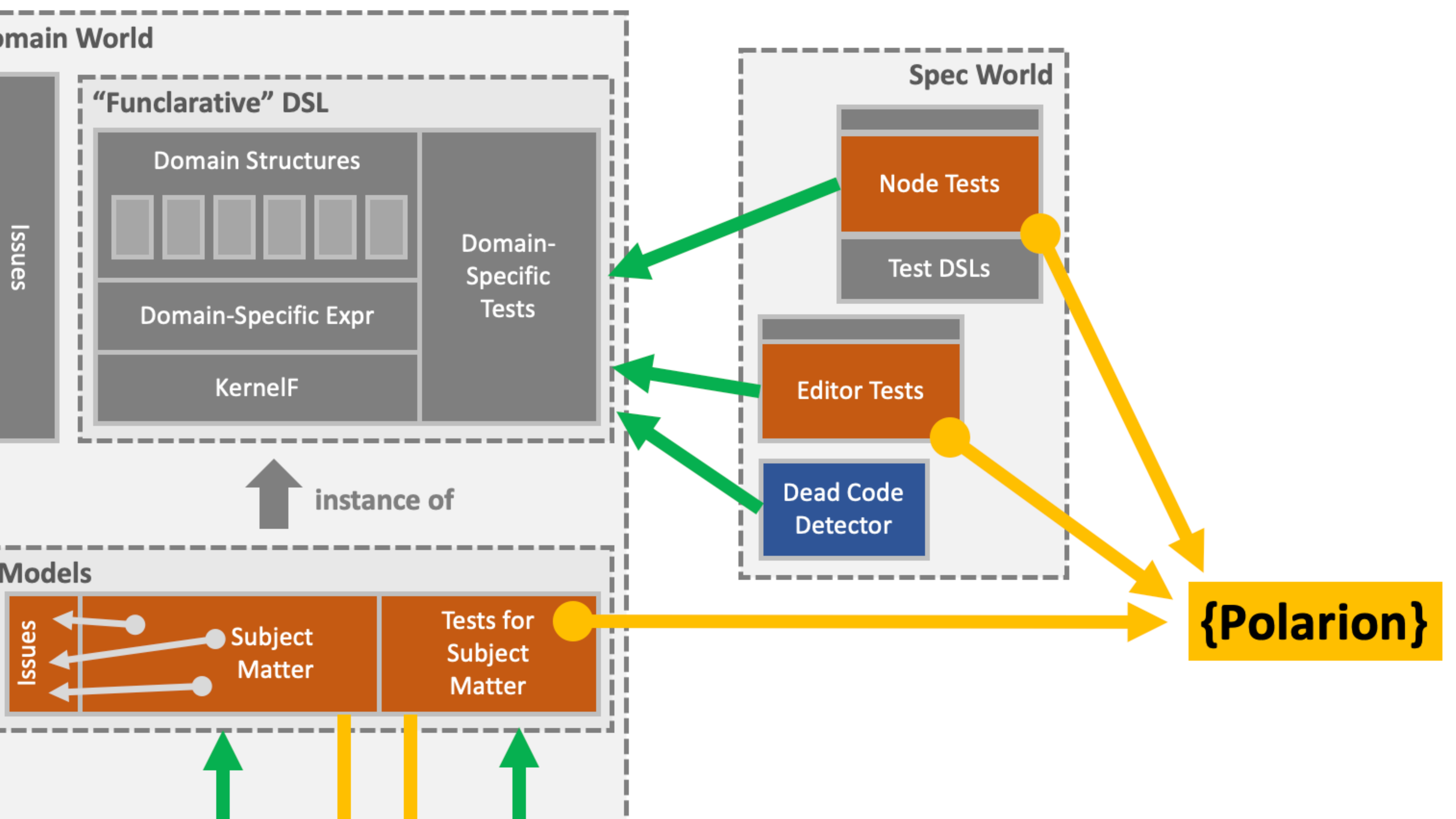

Kernel
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
Interactive, UI-based simulator
Interpreter
{Java}
G
G
{Existing Runtimes}

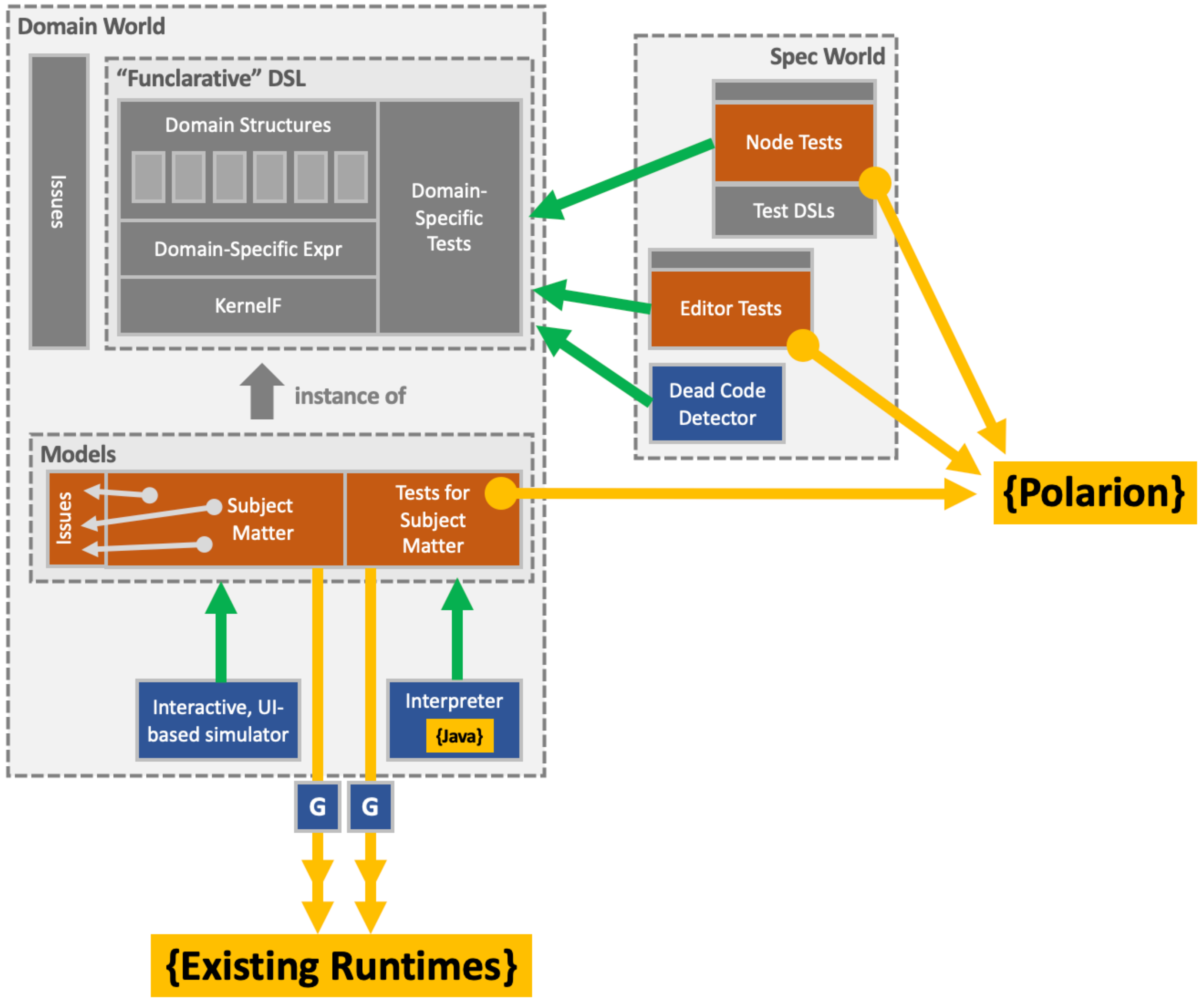

Domain World
Issues
"Funclarative" DSL
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
Node Tests
Editor Tests
instance of
Models
Issues
Subject Matter
Tests for Subject Matter

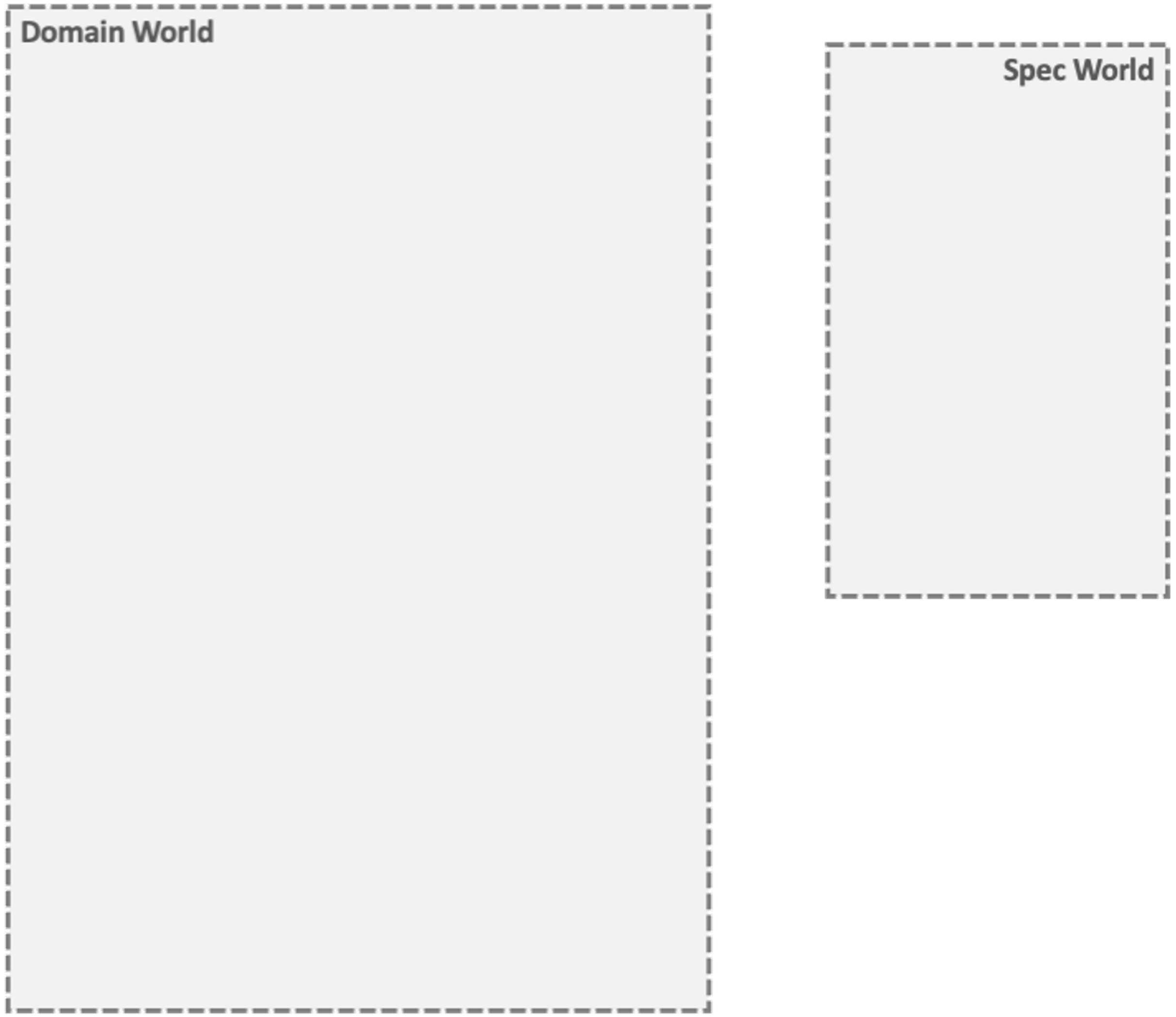

Domain World
"Funclarative" DSL
Issues
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
Node Tests
Test DSLs
Editor Tests
instance of
Models
Issues
Subject Matter
Tests for Subject Matter


main World
Issues
"Funclarative" DSL
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
Node Tests
Test DSLs
Editor Tests
{Polarion}
instance of
Models
Issues
Subject Matter
Tests for Subject Matter


main World
"Funclarative" DSL
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
Issues
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
Node Tests
Test DSLs
Editor Tests
{Polarion}

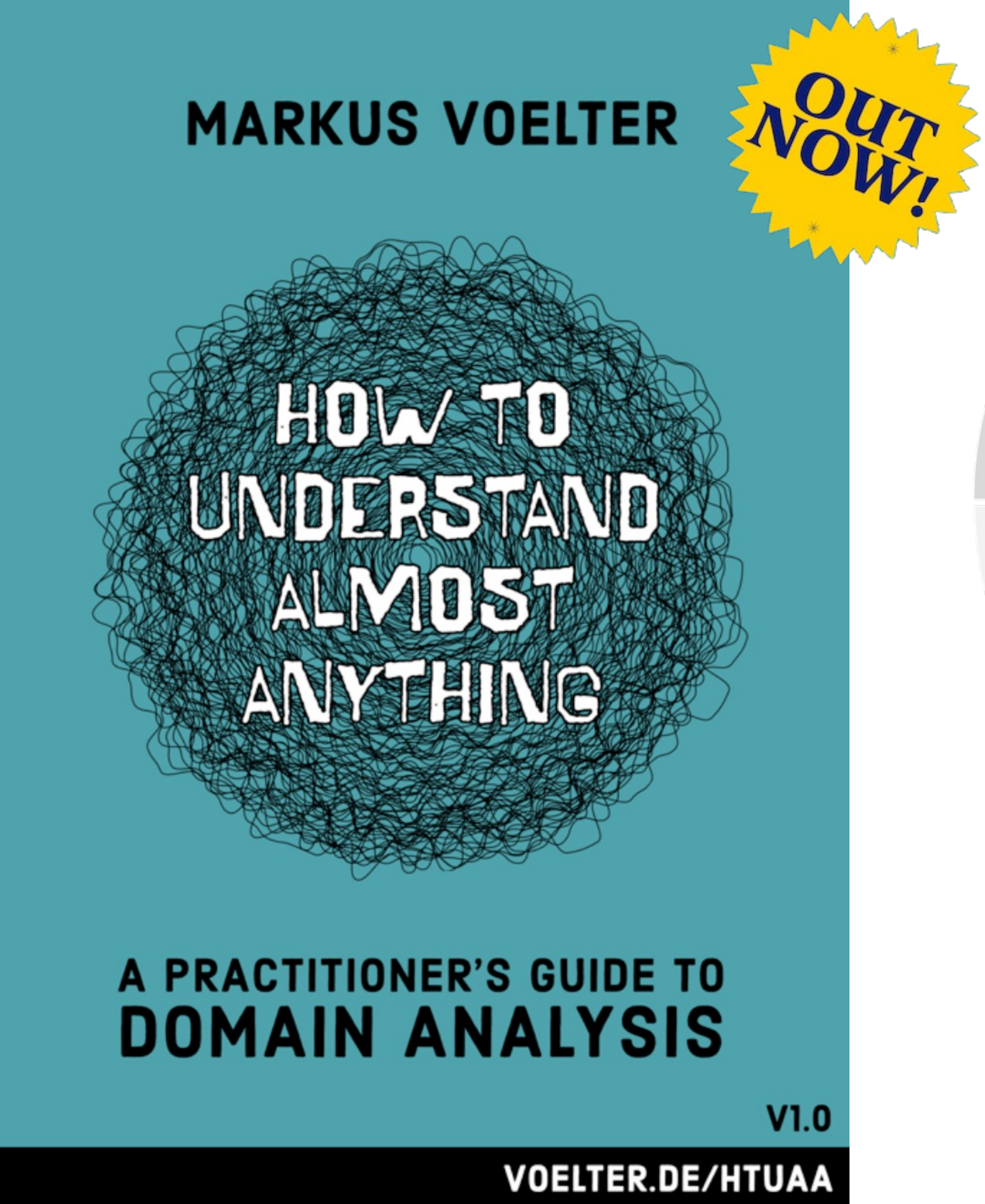

omain World
"Funclarative" DSL
Issues
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
Spec World
Node Tests
Test DSLs
Editor Tests
Dead Code Detector
{Polarion}

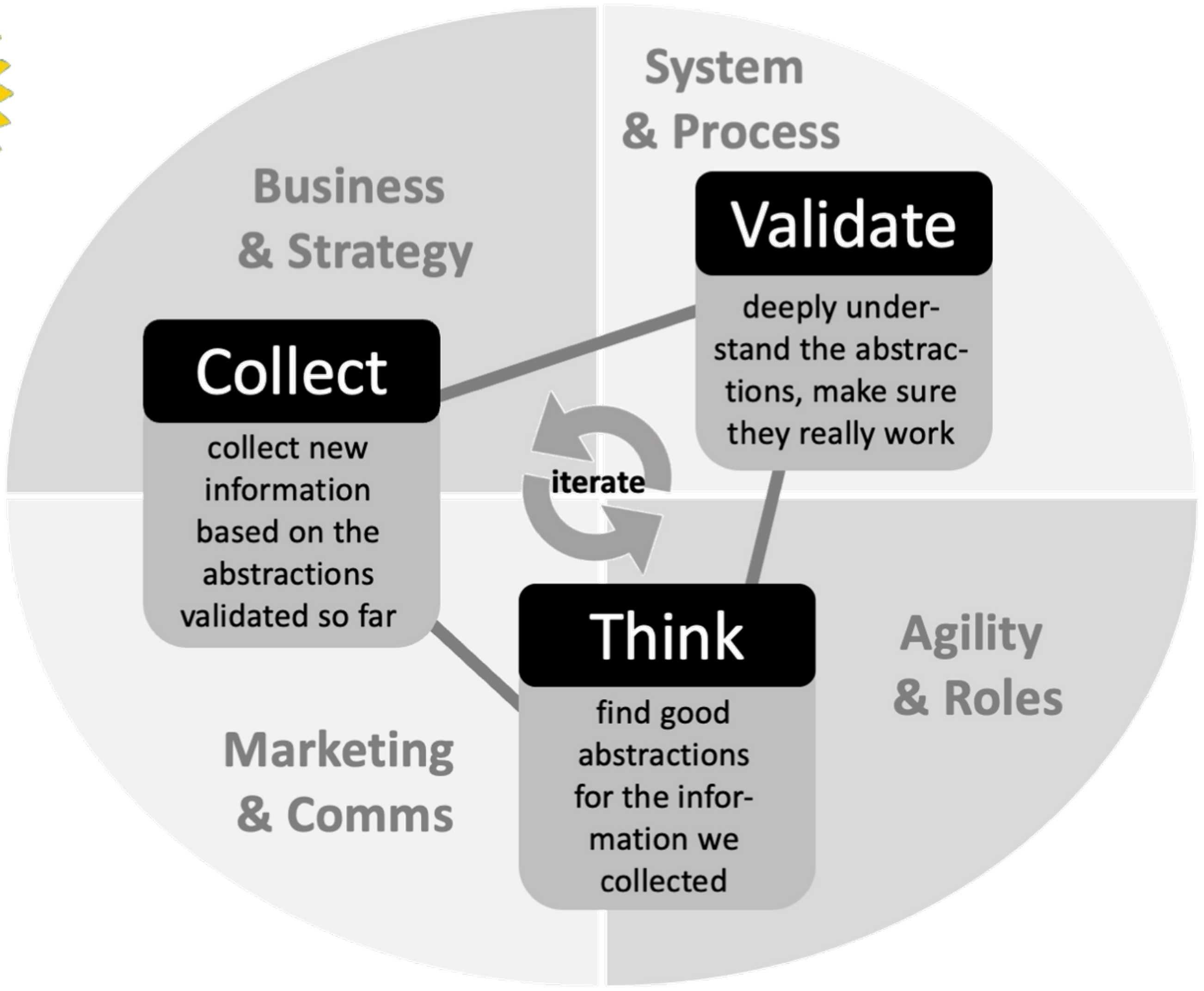

Domain World
"Funclarative" DSL
Issues
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
Interactive, UI-based simulator
Interpreter
{Java}
G
G
{Existing Runtimes}
Spec World
Node Tests
Test DSLs
Editor Tests
Dead Code Detector
{Polarion}


Domain World
Spec World

WE INTERRUPT THIS PROGRAM FOR A

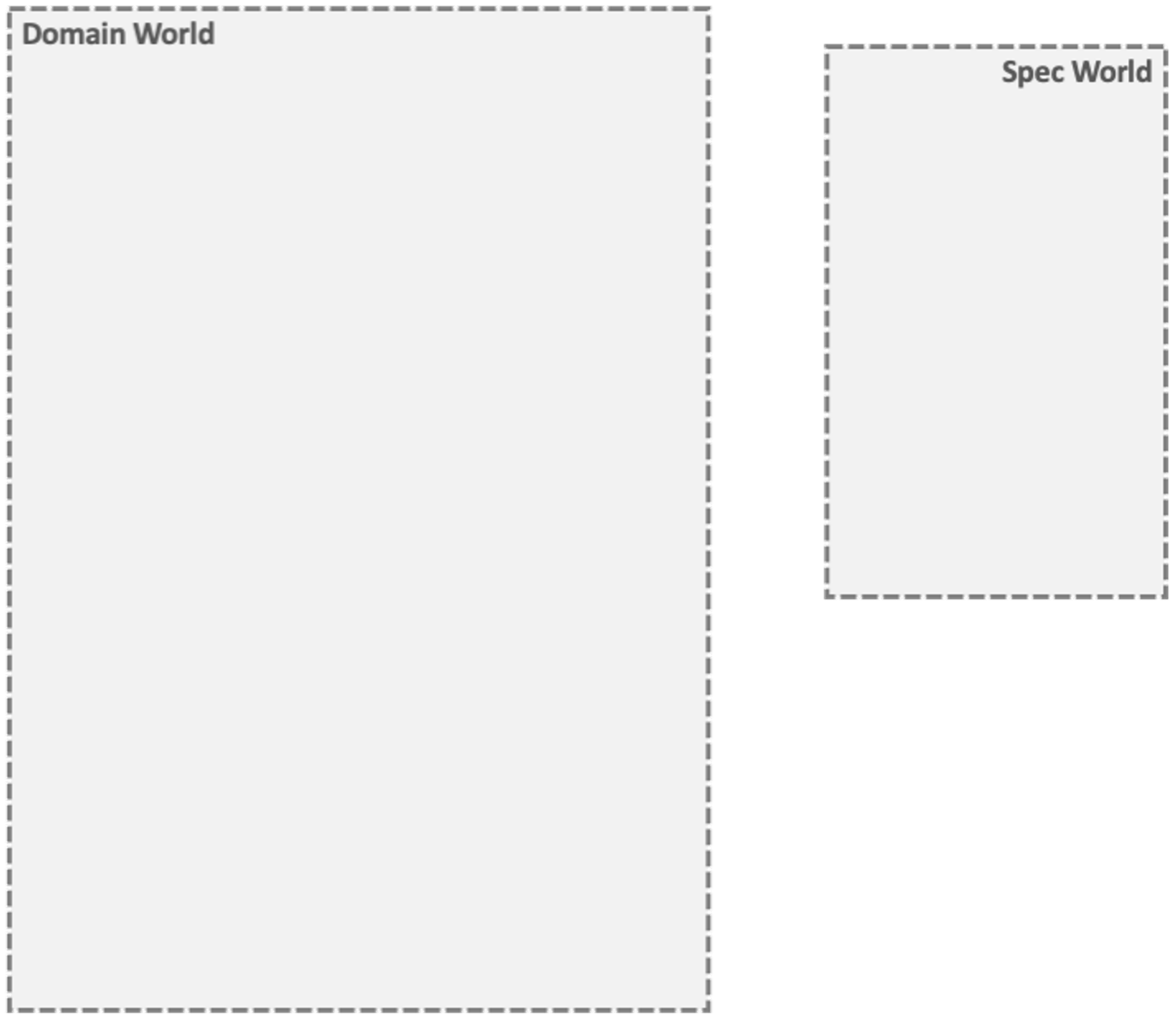
COMMERCIAL BREAK

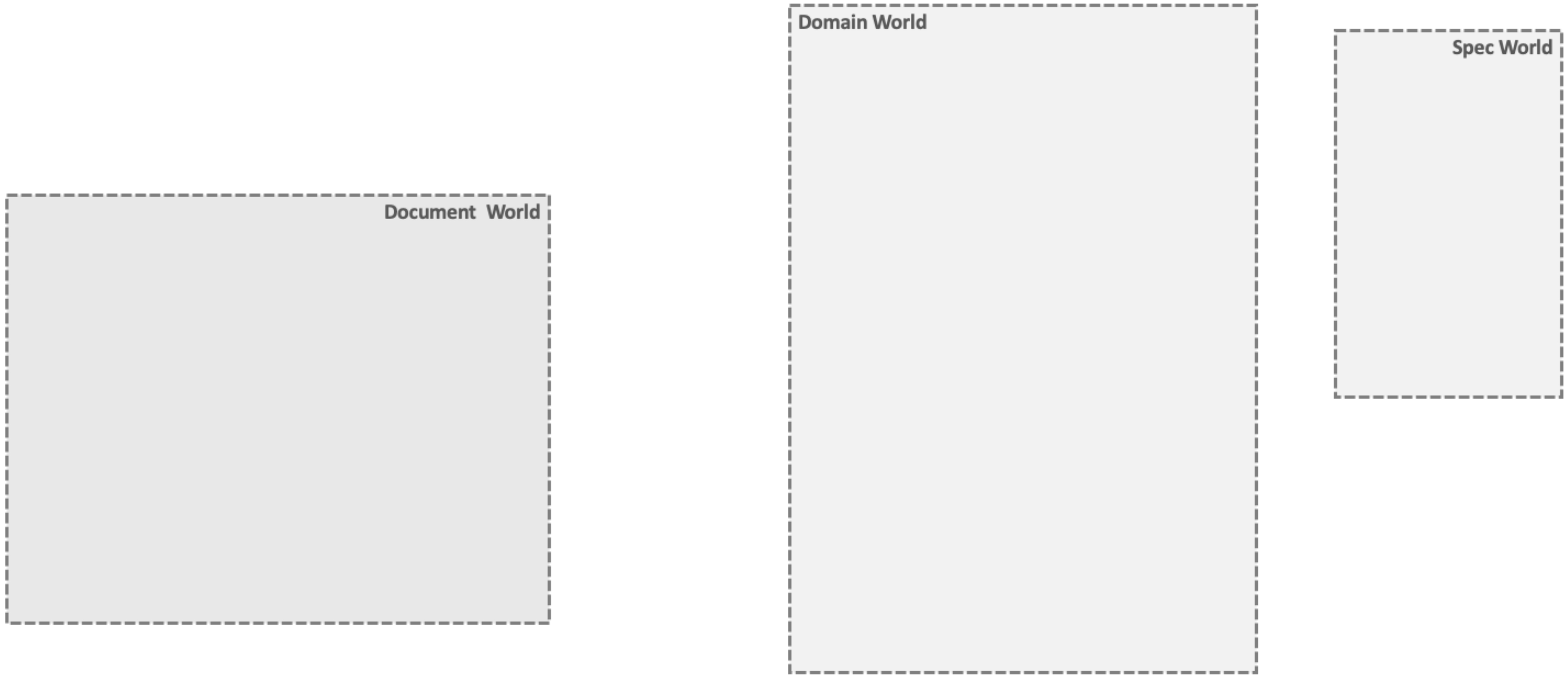

MARKUS VOELTER
OUT NOW!
HOW TO UNDERSTAND ALMOST ANYTHING
A PRACTITIONER'S GUIDE TO DOMAIN ANALYSIS
V1.0
VOELTER.DE/HTUAA

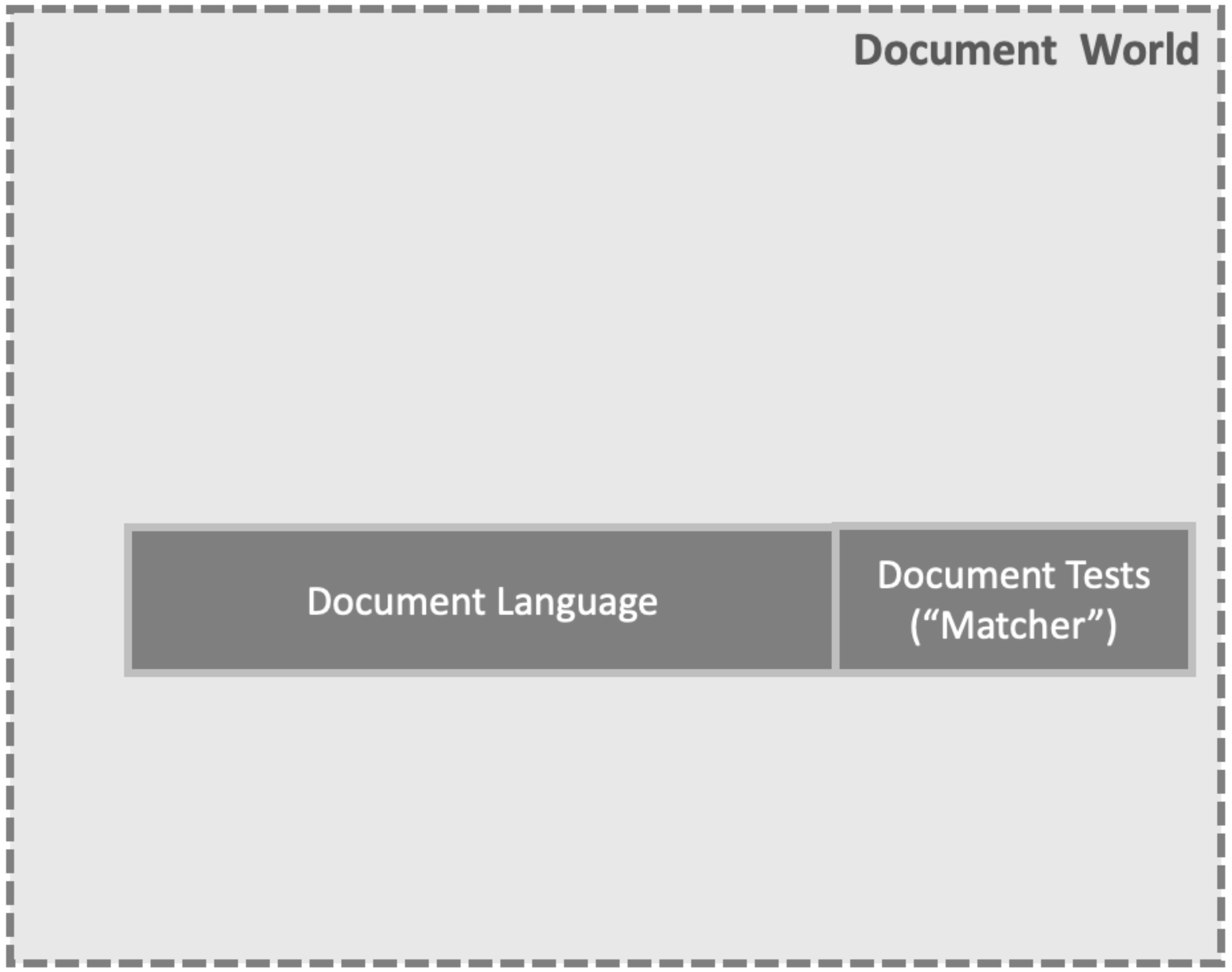

System & Process
Business & Strategy
Validate
deeply understand the abstractions, make sure they really work
Collect
collect new information based on the abstractions validated so far
iterate
Think
find good abstractions for the information we collected
Agility & Roles
Marketing & Comms

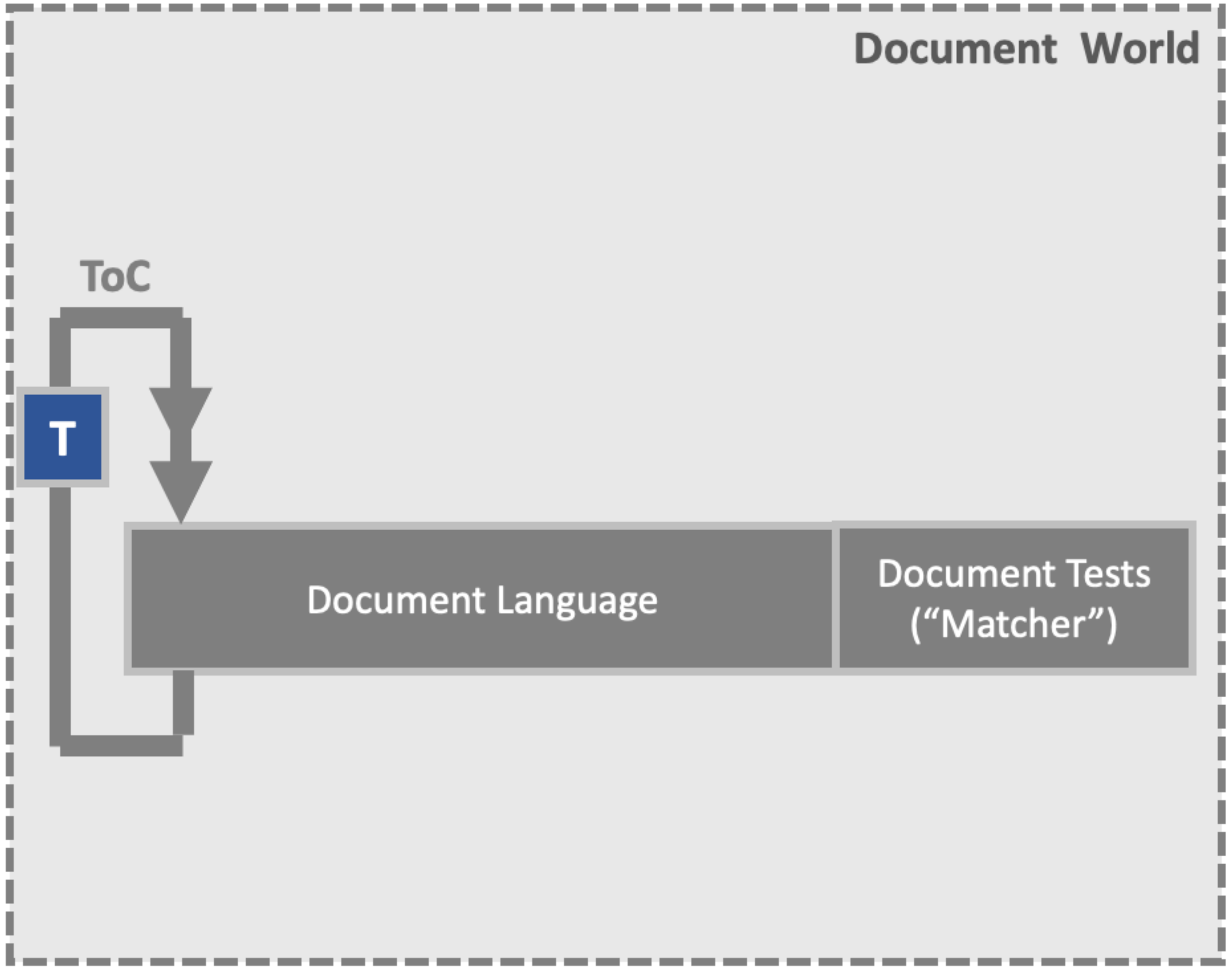
voelter.de/htuaa

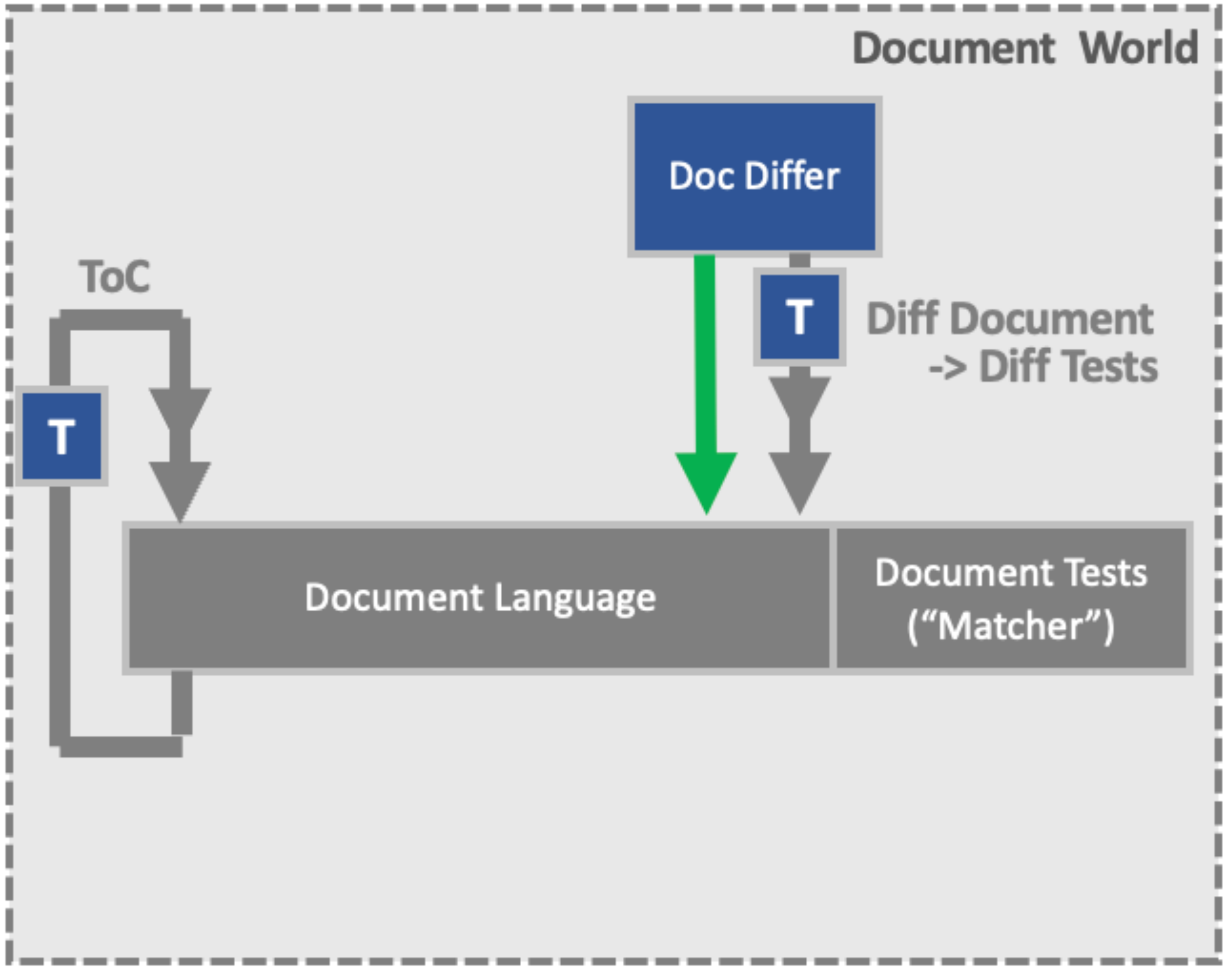
AND NOW BACK TO OUR REGULARLY SCHEDULED PROGRAMMING

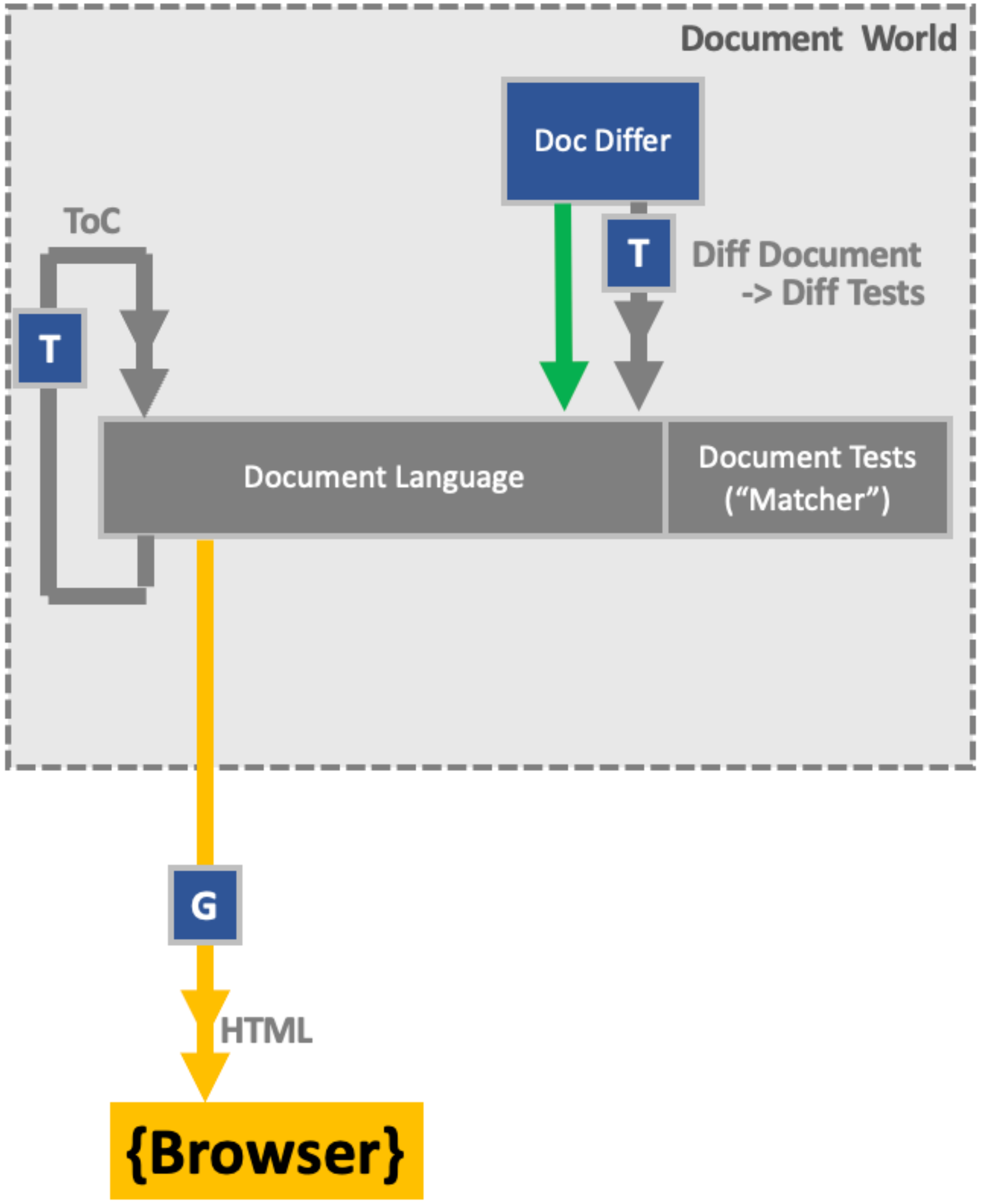

Domain World
Spec World

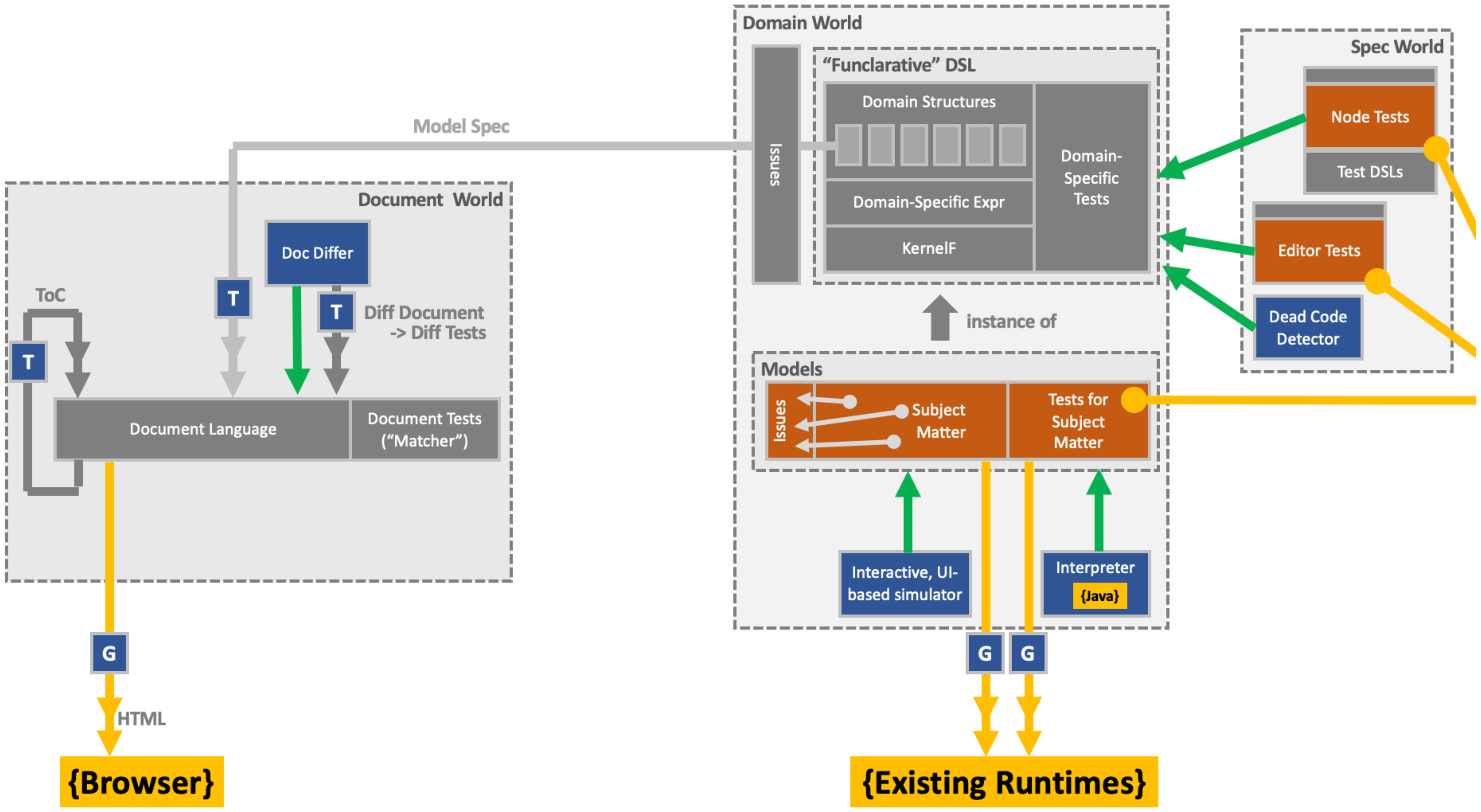

Document World
Domain World
Spec World


Document World
Document Language
Document Tests
("Matcher")

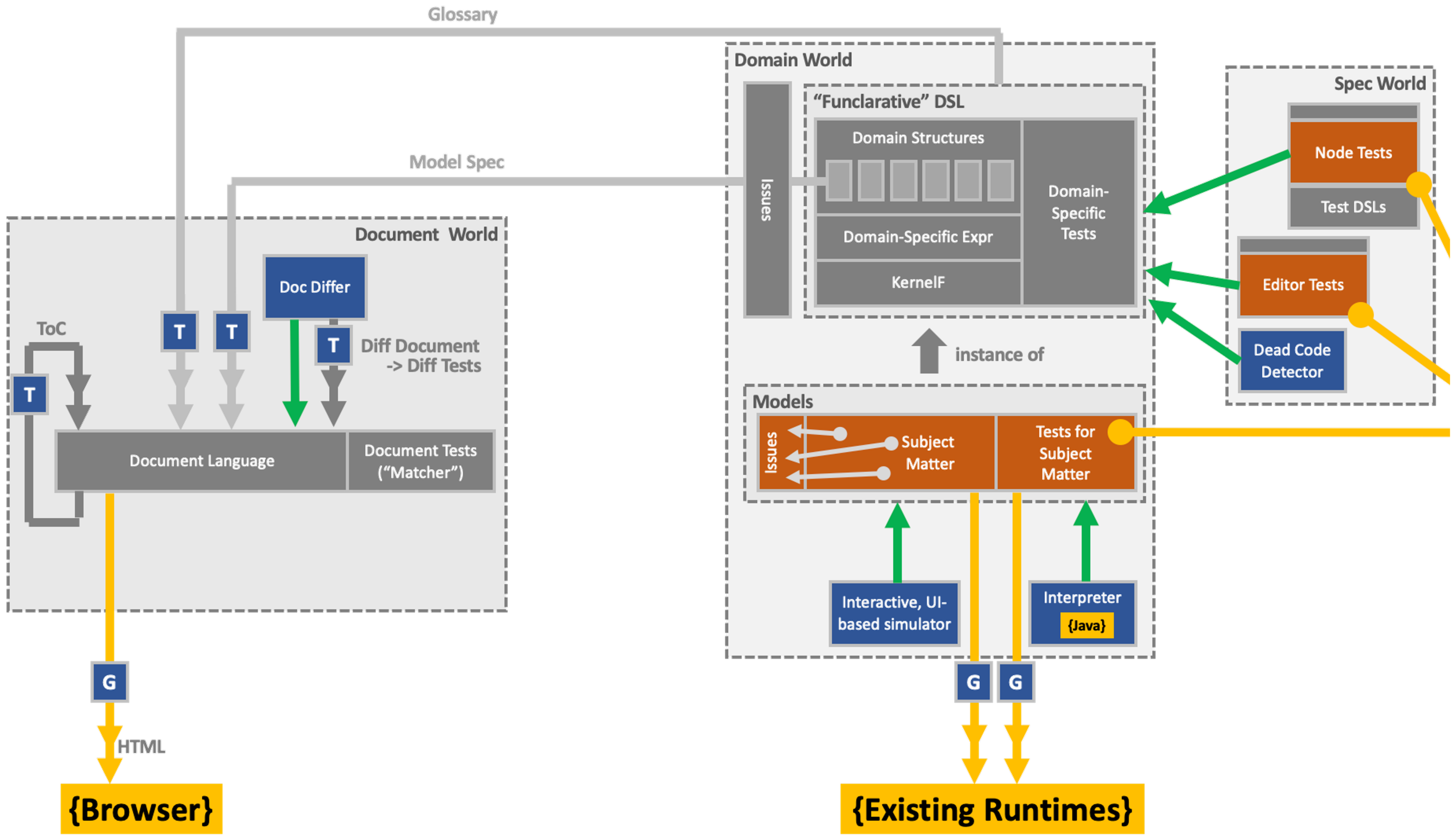

Document World
ToC
T
Document Language
Document Tests
("Matcher")


Document World
Doc Differ
ToC
T
T
Diff Document
-> Diff Tests
Document Language
Document Tests
("Matcher")

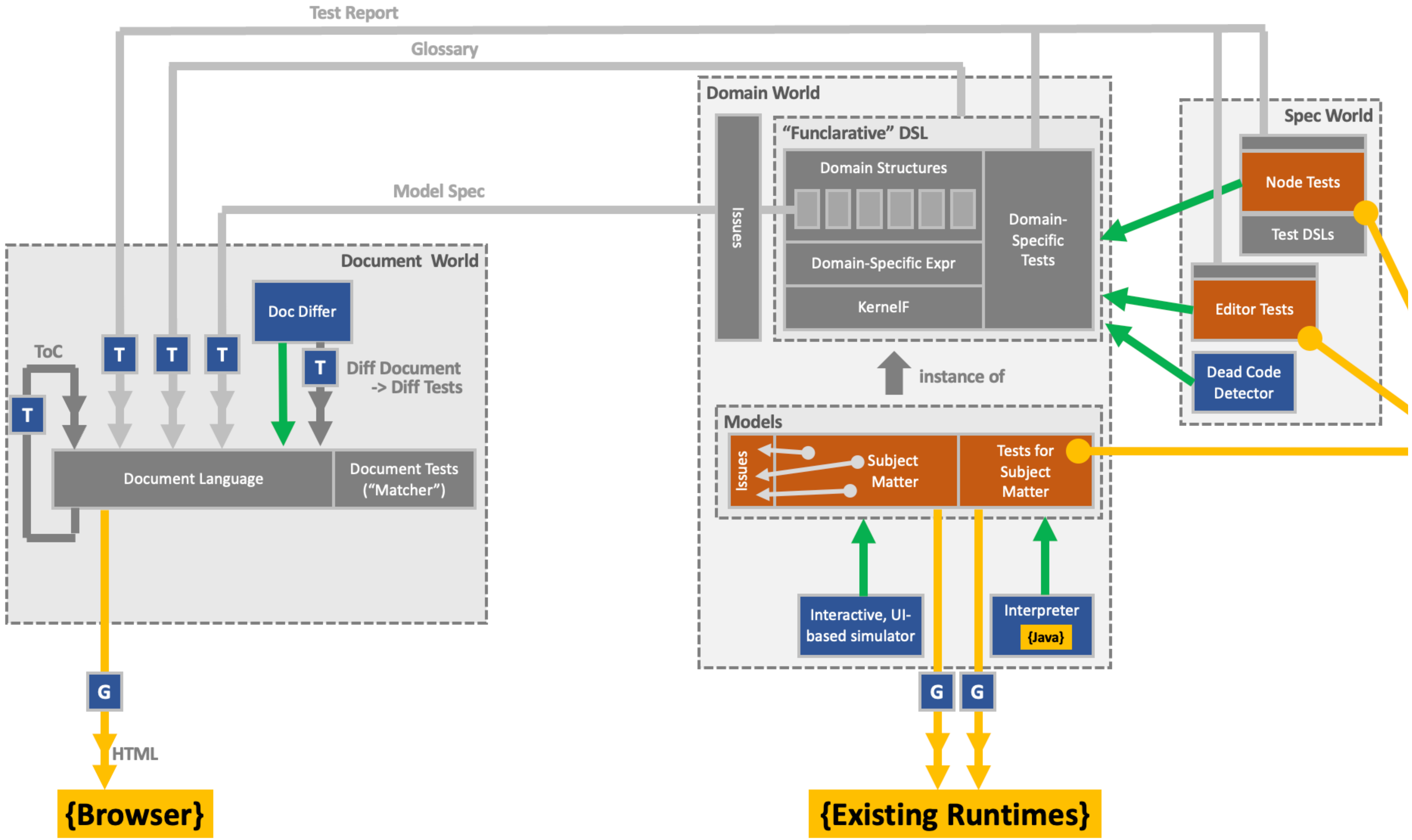

Document World
Doc Differ
ToC
T
T
Diff Document -> Diff Tests
Document Language
Document Tests ("Matcher")
G
HTML
{Browser}

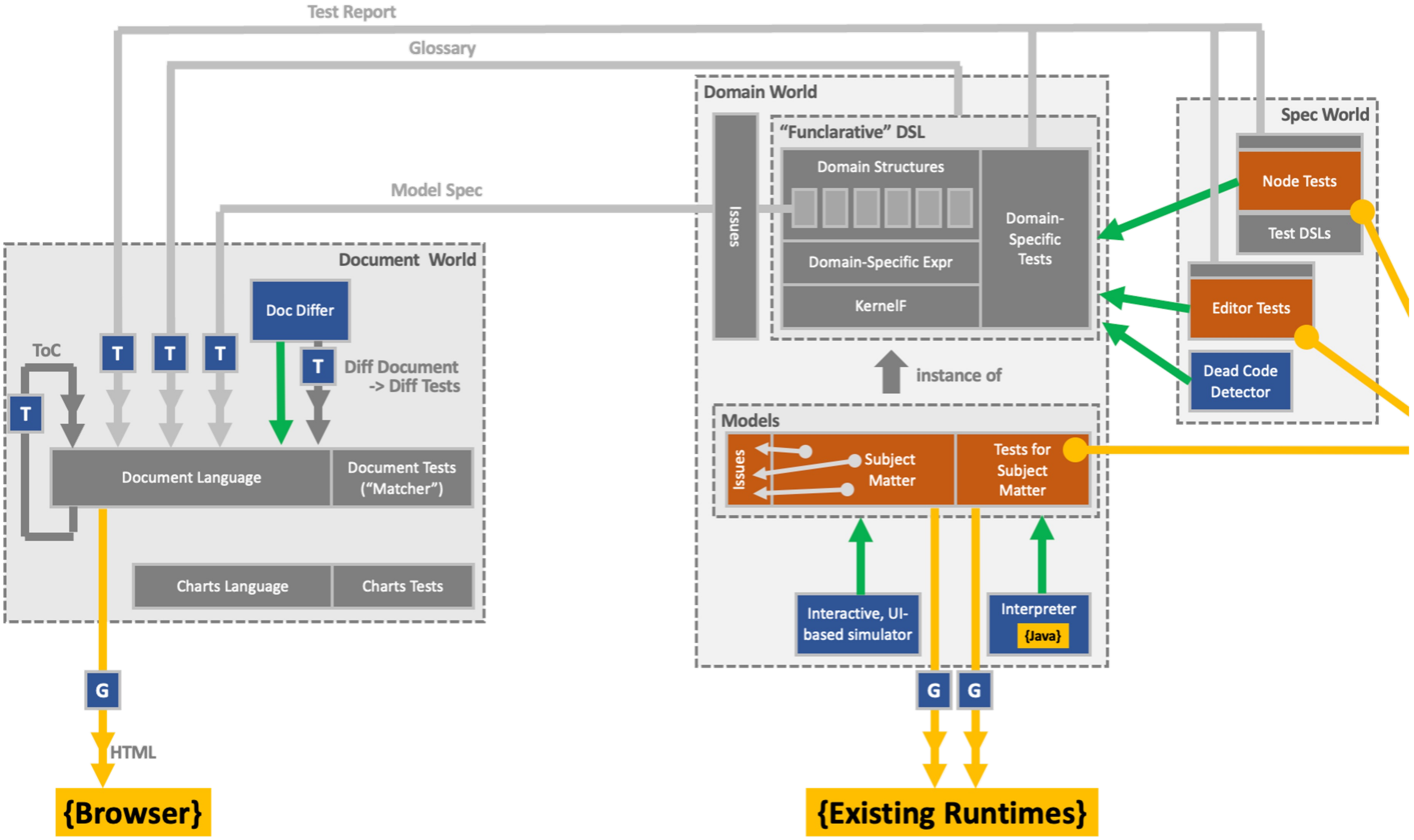


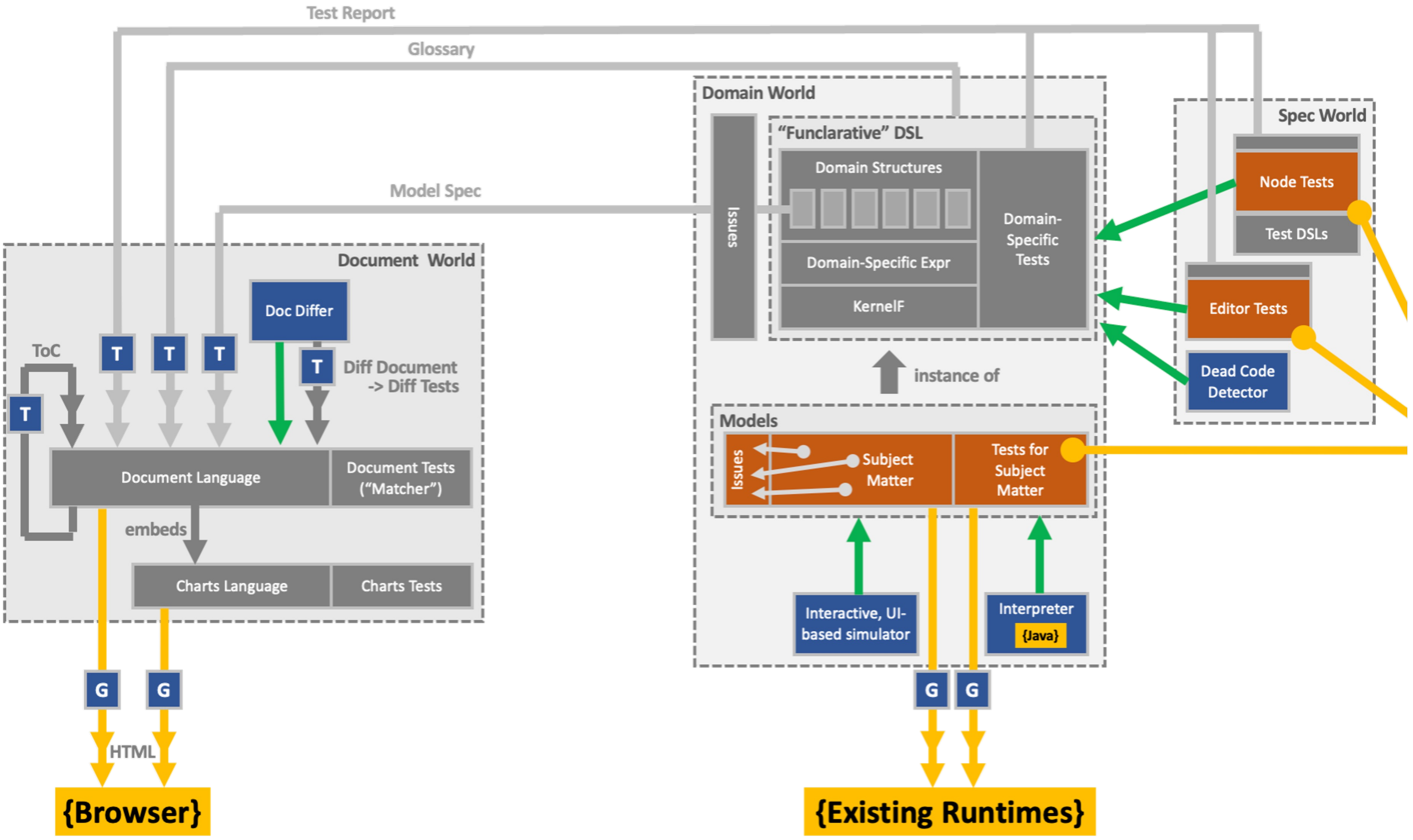
Model Spec
Domain World
"Funclarative" DSL
Domain Structures
Issues
Domain-Specific Expr
KernelF
Domain-Specific Tests
Spec World
Node Tests
Test DSLs
Editor Tests
Dead Code Detector
Document World
Doc Differ
ToC
T
T
T
Diff Document -> Diff Tests
Document Language
Document Tests ("Matcher")
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
Interactive, UI-based simulator
Interpreter
{Java}
G
G
G
HTML
{Browser}
{Existing Runtimes}

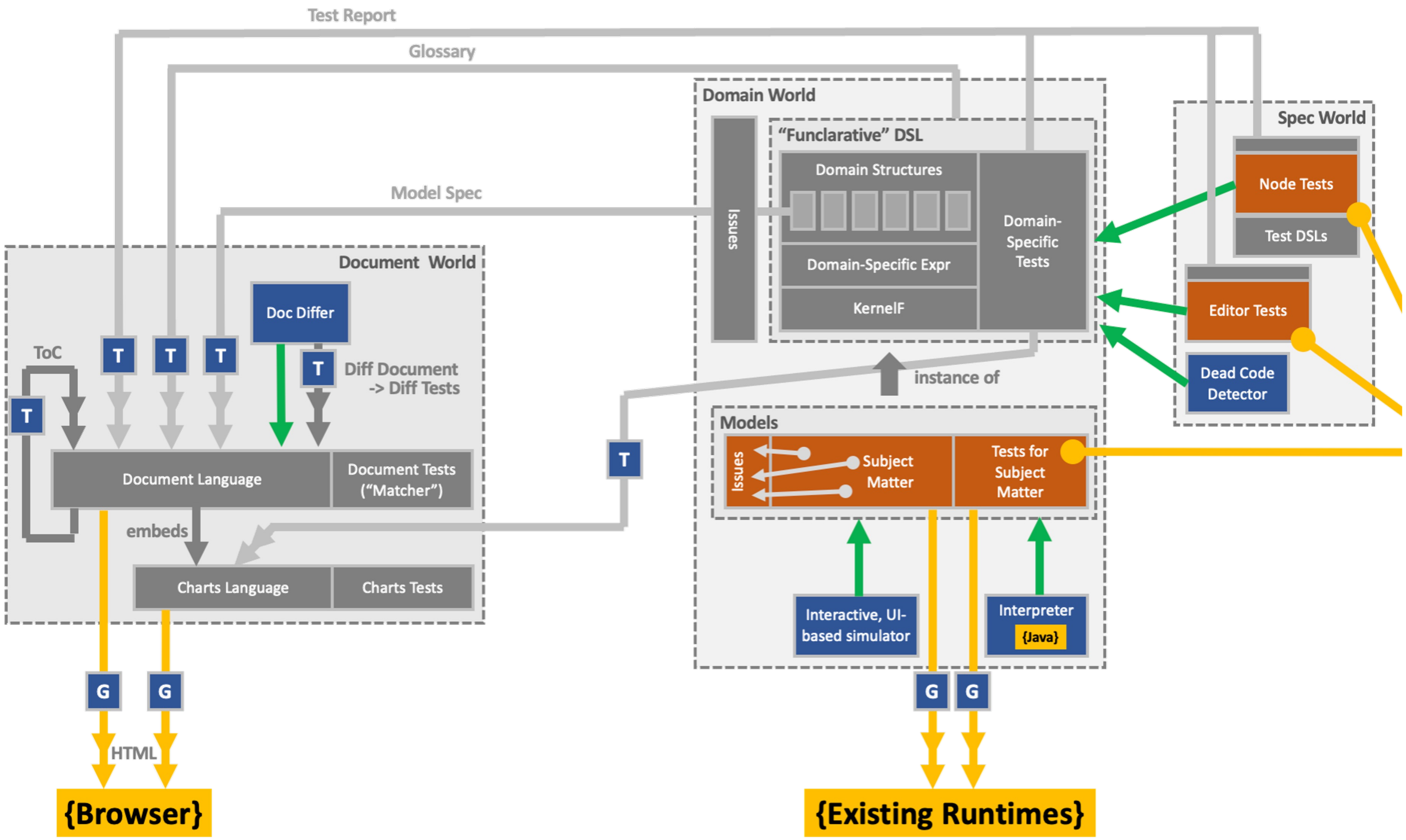


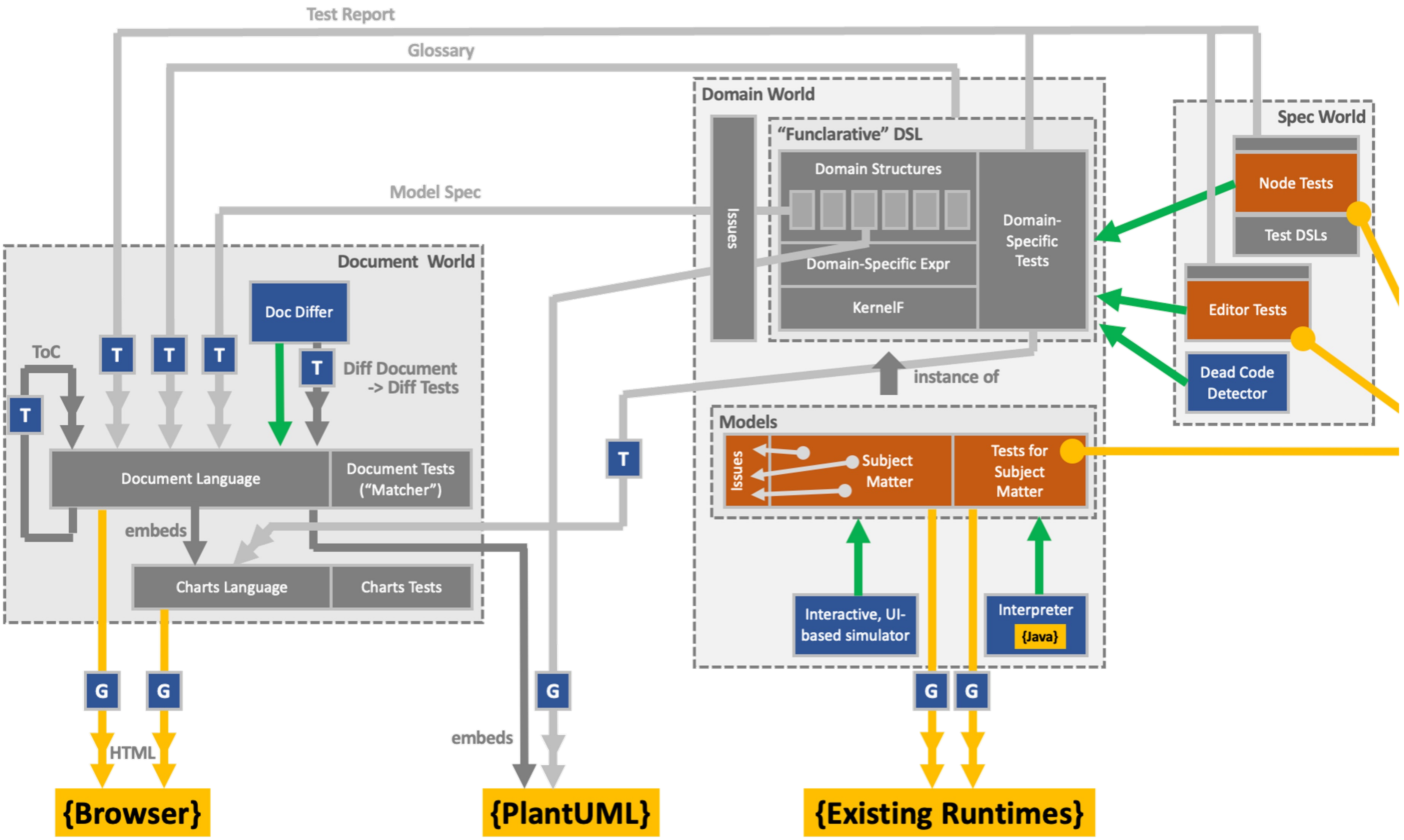
Glossary
Model Spec
Document World
Doc Differ
T
T
T
T
ToC
Diff Document
-> Diff Tests
Document Language
Document Tests
("Matcher")
G
HTML
{Browser}
Domain World
"Funclarative" DSL
Issues
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
Interactive, UI-based simulator
Interpreter
{Java}
G
G
{Existing Runtimes}
Spec World
Node Tests
Test DSLs
Editor Tests
Dead Code Detector

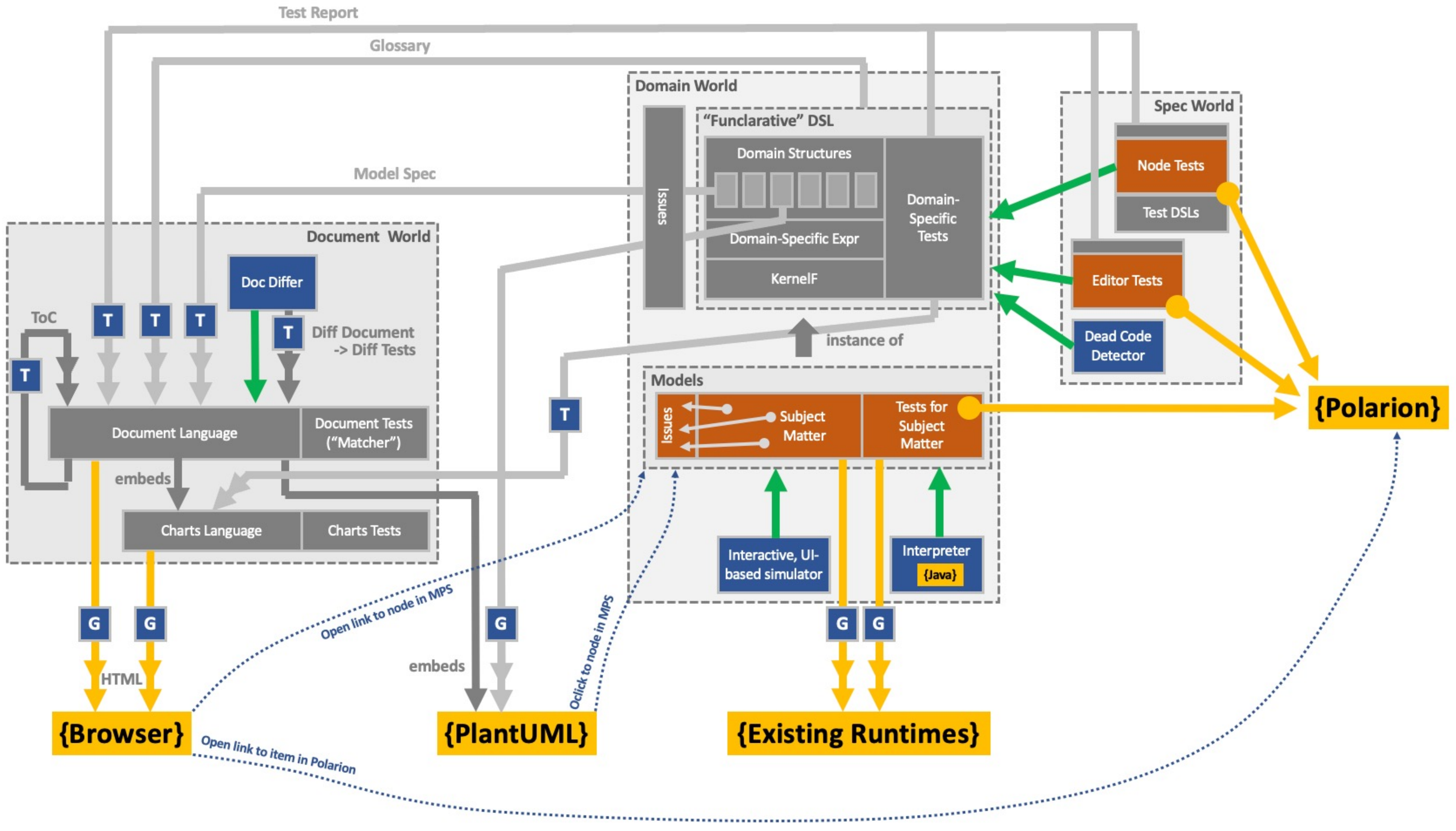

Test Report
Glossary
Model Spec
Domain World
"Funclarative" DSL
Issues
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
Interactive, UI-based simulator
Interpreter
{Java}
Spec World
Node Tests
Test DSLs
Editor Tests
Dead Code Detector
Document World
Doc Differ
ToC
T
Diff Document -> Diff Tests
Document Language
Document Tests ("Matcher")
G
HTML
{Browser}
{Existing Runtimes}


Test Report
Glossary
Model Spec
Document World
Doc Differ
T
ToC
Diff Document
-> Diff Tests
Document Language
Document Tests
("Matcher")
Charts Language
Charts Tests
G
HTML
{Browser}
Domain World
"Funclarative" DSL
Issues
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
instance of
Models
Subject Matter
Tests for Subject Matter
Interactive, UI-based simulator
Interpreter
{Java}
{Existing Runtimes}
Spec World
Node Tests
Test DSLs
Editor Tests
Dead Code Detector

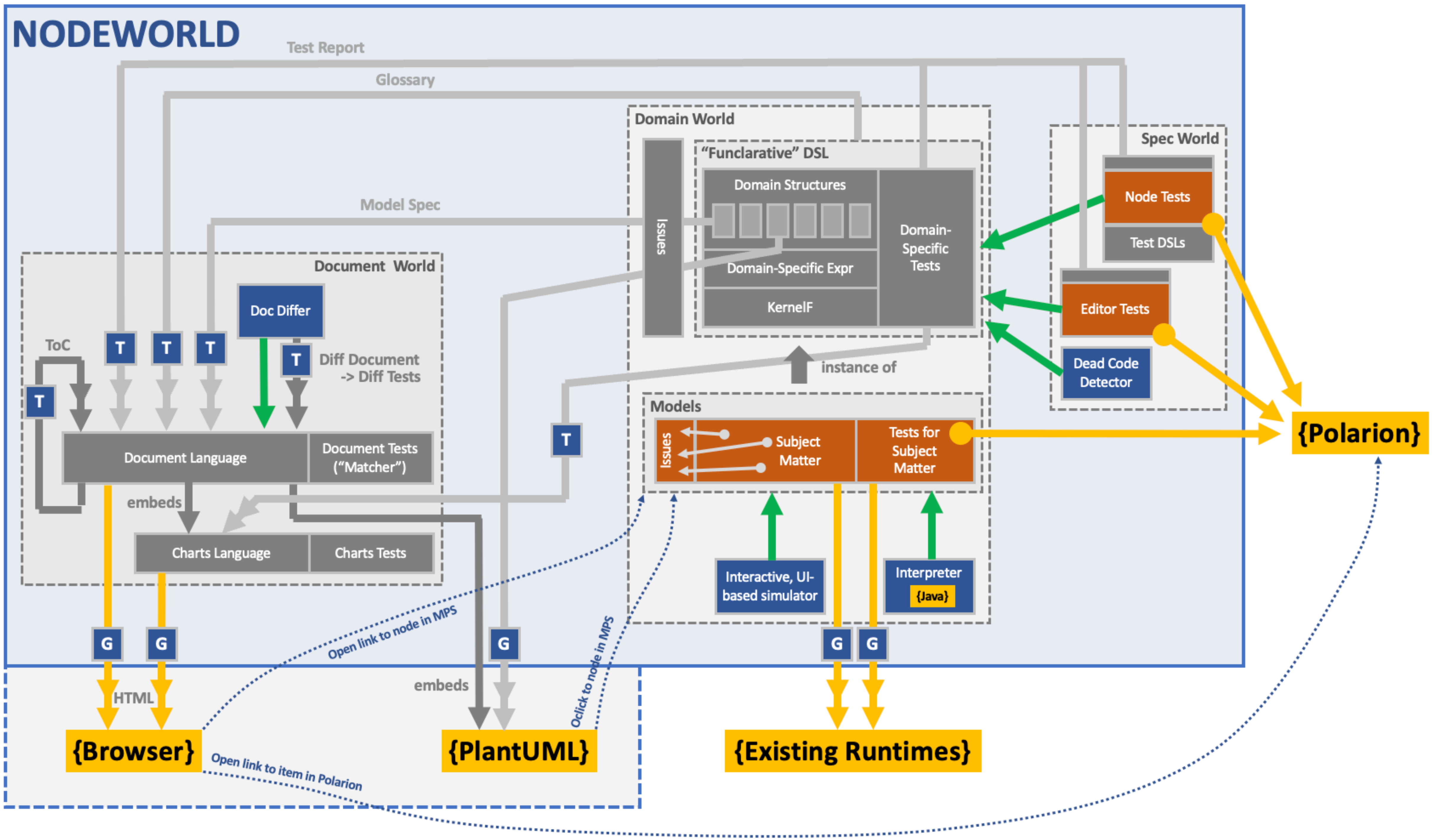

Test Report
Glossary
Model Spec
Domain World
"Funclarative" DSL
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
Issues
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
Interactive, UI-based simulator
Interpreter
{Java}
Spec World
Node Tests
Test DSLs
Editor Tests
Dead Code Detector
Document World
Doc Differ
T
T
T
T
Diff Document -> Diff Tests
ToC
T
Document Language
Document Tests ("Matcher")
embeds
Charts Language
Charts Tests
G
G
HTML
{Browser}
G
G
{Existing Runtimes}


Test Report
Glossary
Model Spec
Domain World
"Funclarative" DSL
Domain Structures
Issues
Domain-Specific Expr
KernelF
Domain-Specific Tests
Spec World
Node Tests
Test DSLs
Editor Tests
Dead Code Detector
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
Interactive, UI-based simulator
Interpreter
{Java}
Document World
Doc Differ
ToC
T
Diff Document -> Diff Tests
Document Language
Document Tests ("Matcher")
embeds
Charts Language
Charts Tests
G
HTML
{Browser}
{Existing Runtimes}


Test Report
Glossary
Model Spec
Domain World
"Funclarative" DSL
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
Issues
Spec World
Node Tests
Test DSLs
Editor Tests
Dead Code Detector
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
Interactive, UI-based simulator
Interpreter
{Java}
Document World
Doc Differ
Diff Document -> Diff Tests
ToC
T
Document Language
Document Tests ("Matcher")
embeds
Charts Language
Charts Tests
G
HTML
embeds
{Browser}
{PlantUML}
{Existing Runtimes}



Test Report
Glossary
Model Spec
Domain World
"Funclarative" DSL
Issues
Domain Structures
Domain-Specific Expr
KernelF
Domain-Specific Tests
Spec World
Node Tests
Test DSLs
Editor Tests
Dead Code Detector
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
Interactive, UI-based simulator
Interpreter
{Java}
Document World
Doc Differ
ToC
T
Diff Document -> Diff Tests
Document Language
Document Tests ("Matcher")
embeds
Charts Language
Charts Tests
G
HTML
embeds
{Browser}
{PlantUML}
{Existing Runtimes}
{Polarion}
Open link to node in MPS
Oclick to node in MPS
Open link to item in Polarion


NODEWORLD
Test Report
Glossary
Model Spec
Domain World
"Funclarative" DSL
Domain Structures
Issues
Domain-Specific Expr
KernelF
Domain-Specific Tests
Spec World
Node Tests
Test DSLs
Editor Tests
Dead Code Detector
Document World
Doc Differ
ToC
T
Diff Document -> Diff Tests
Document Language
Document Tests ("Matcher")
embeds
Charts Language
Charts Tests
instance of
Models
Issues
Subject Matter
Tests for Subject Matter
Interactive, UI-based simulator
Interpreter
{Java}
G
HTML
embeds
Open link to node in MPS
Oclick to node in MPS
Open link to item in Polarion
{Browser}
{PlantUML}
{Existing Runtimes}
{Polarion}

SO!

Why does MPS have the best turtles?

Because they have lots of tools under their shell

Notation | Validation | Transformation | Interpretation | Customi Actions

Some things aren't turtles

External systems for requirements

Mandated by Process

PlantUML generation is direct text generation

PlantUML is so big, it's not feasible to build lang

Interpreter does not "reshuffle" nodes

Performance – the Java impl is already slow

Wrap Up

If everything’s the same, it’s easy to build {meta*}-stuff

The “things” should also be accessible to the IDE to support static analysis and processing

Having editors and other IDE aspects as part of the the „thingness“ makes tool-building accessible.

Maybe we shouldn‘t „sell“ MPS (and similar systems) by using the term DSL , but terminology that emphasises the sameness and its consequences.

TATWD.