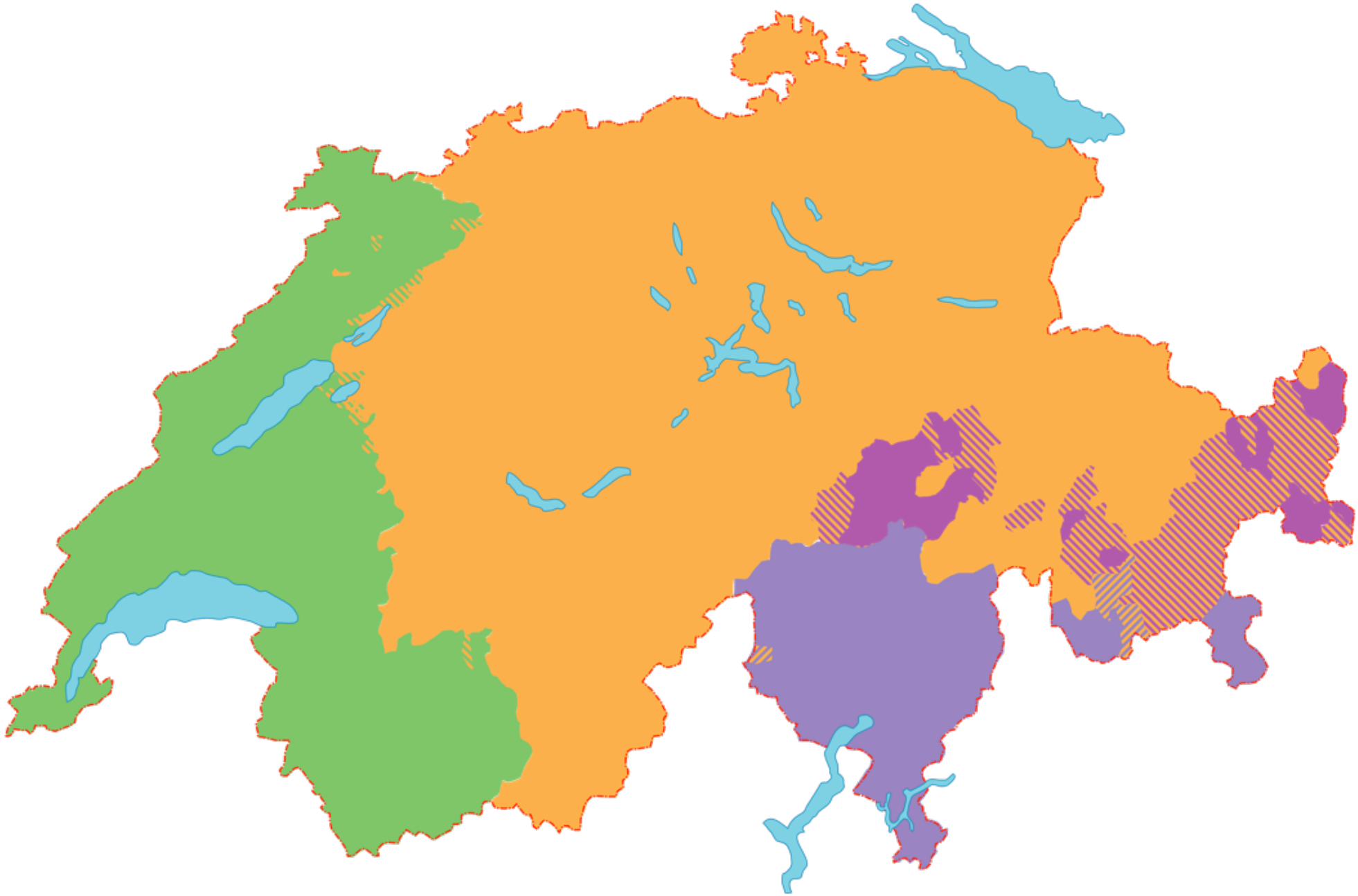
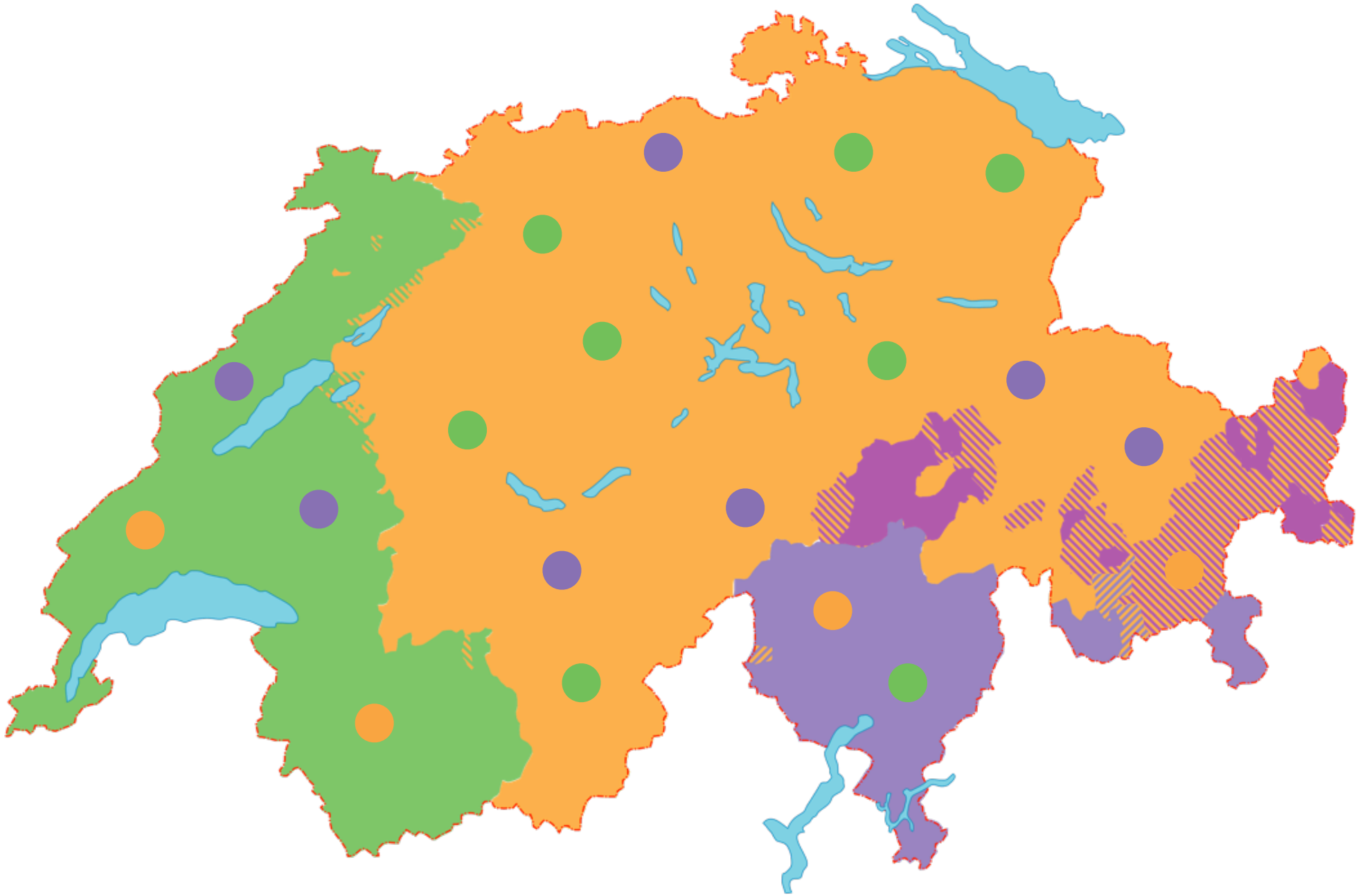
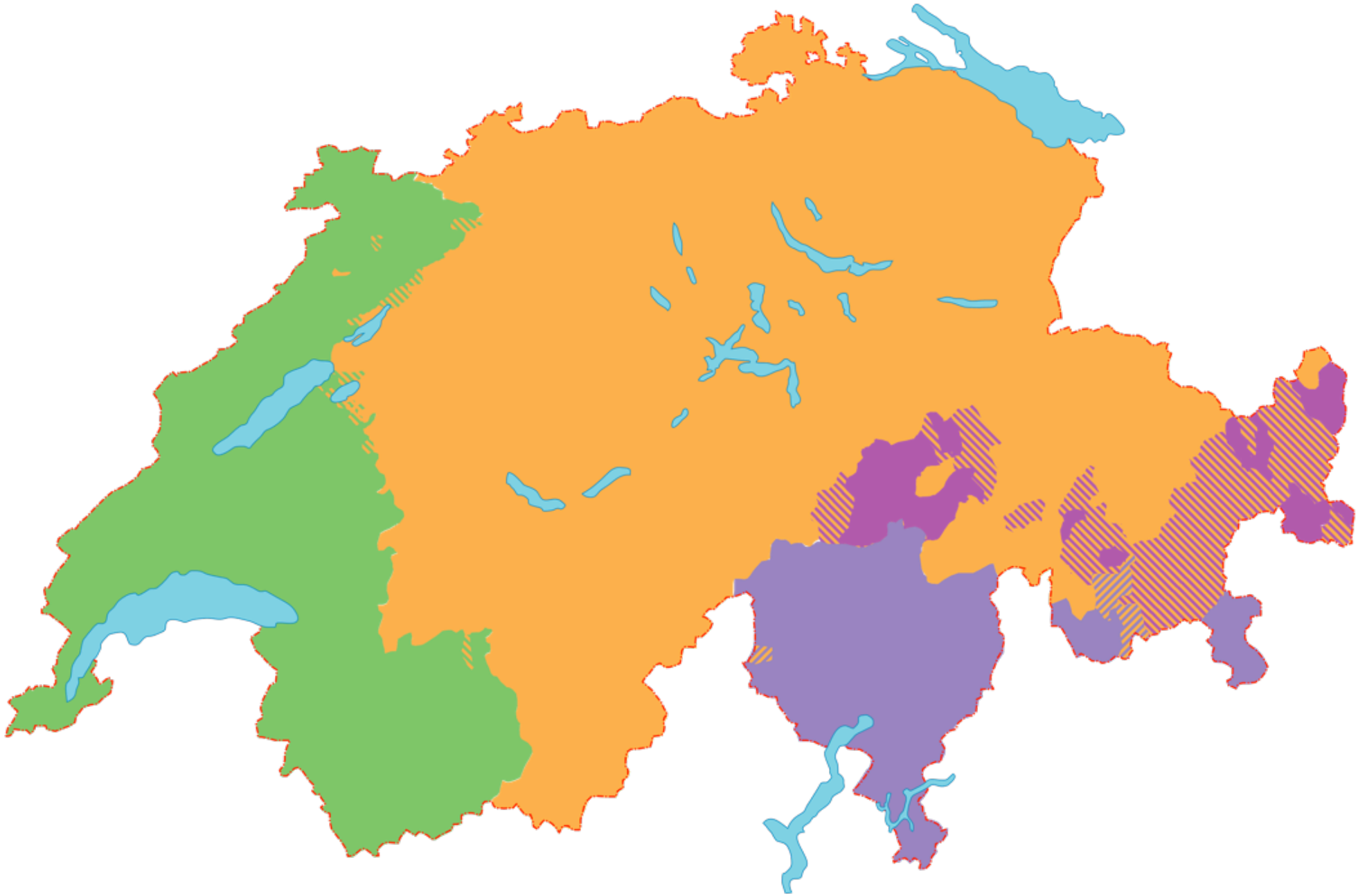
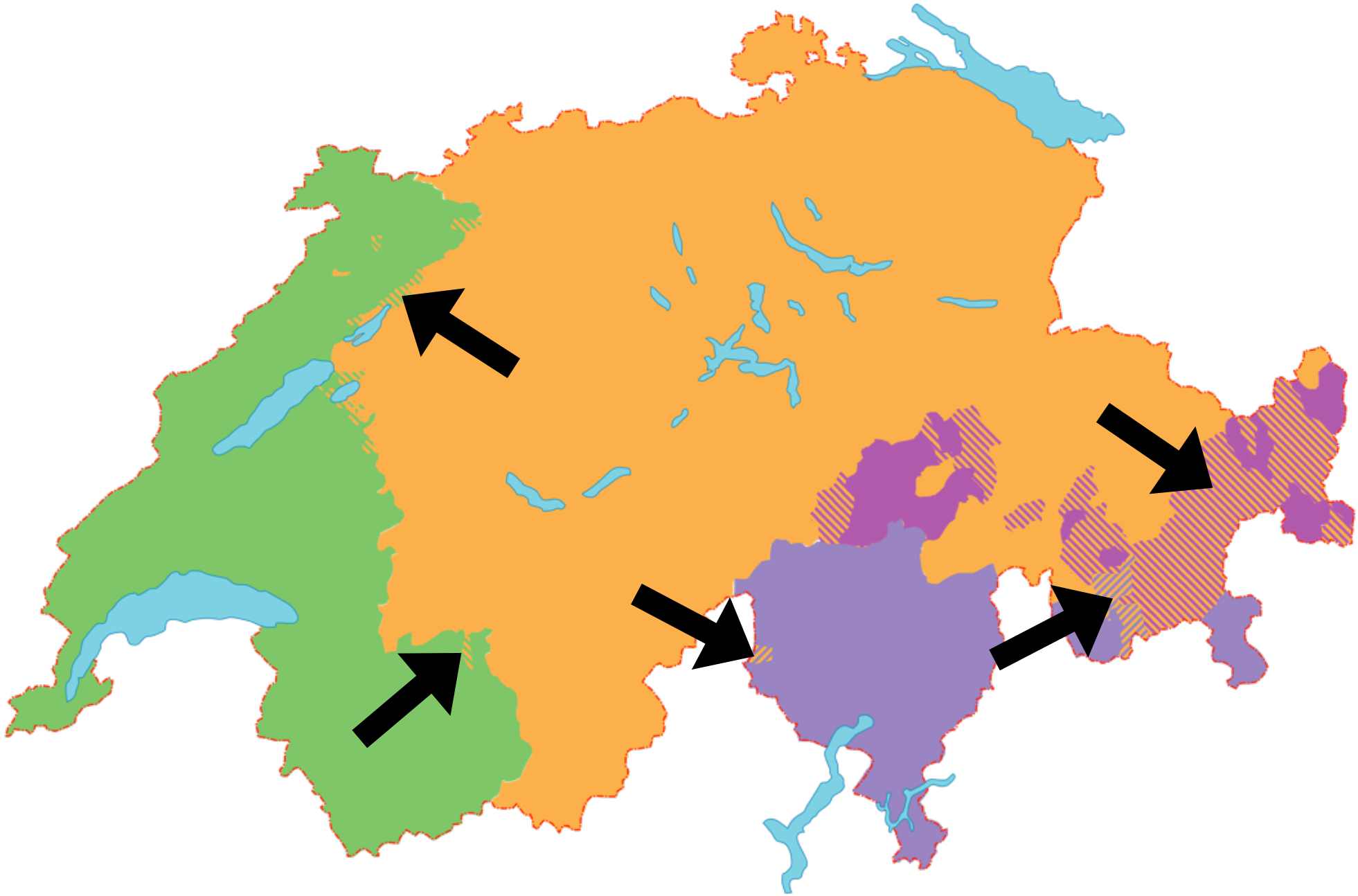

Helvetia






Helvetia

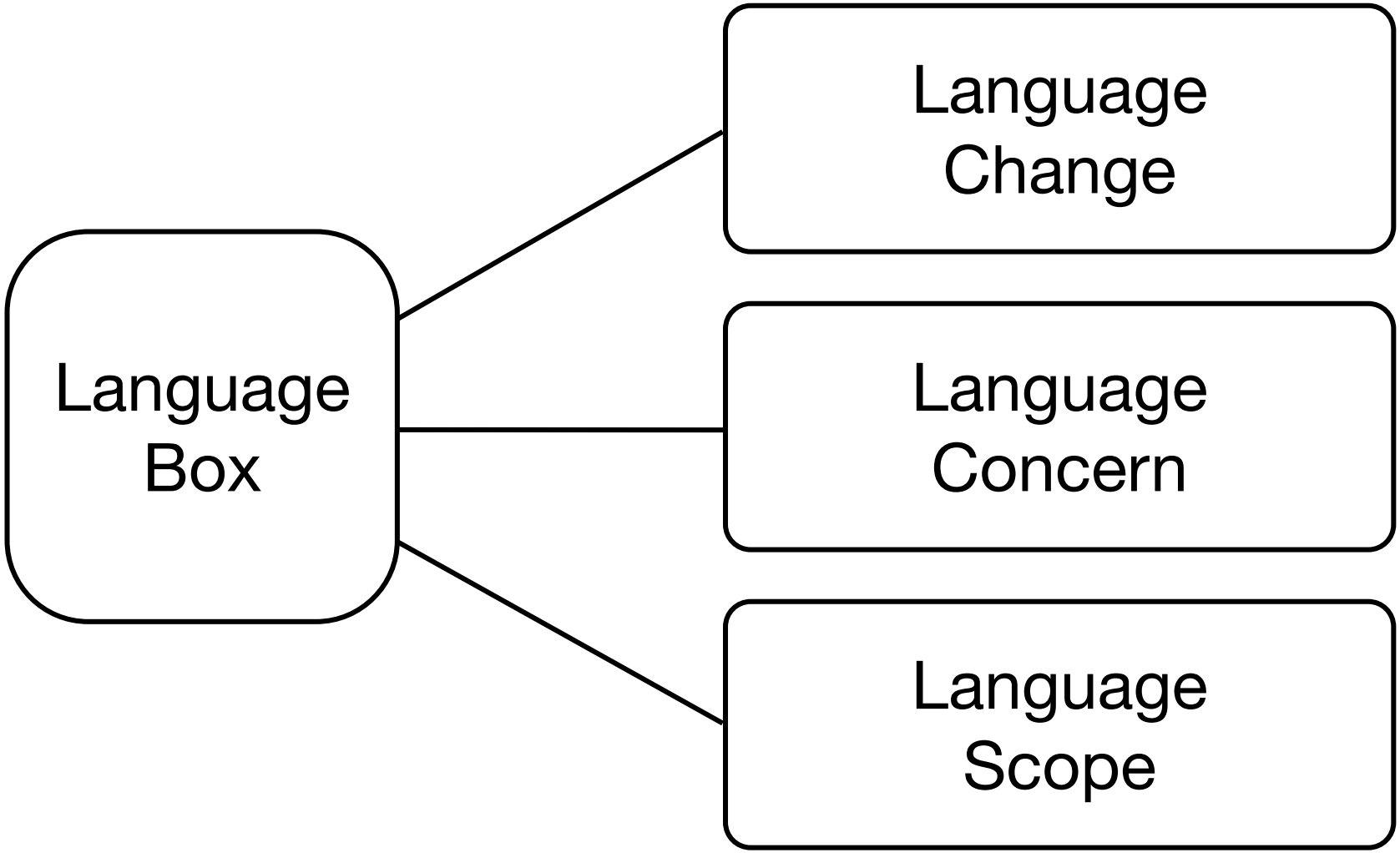




Language Boxes

Tool Infrastructure

Host Environment


Language
Box
Language
Change
Language
Concern
Language
Scope

Demo

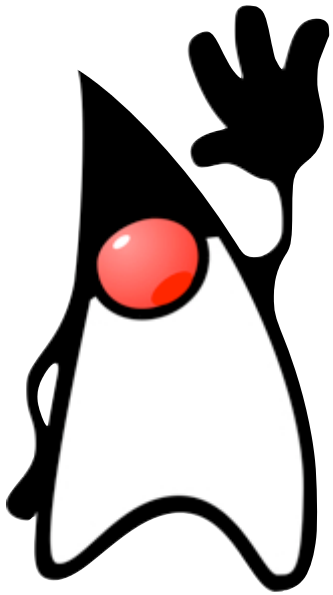
Why Smalltalk?

1. Minimal Syntax
2. Dynamic Semantics
3. Reflective Facilities
4. Homogeneous Language
5. Homogeneous Tools
6. On-the-fly Programming

1. Minimal Syntax
2. Dynamic Semantics
3. Reflective Facilities
4. Homogeneous Language
5. Homogeneous Tools
6. On-the-fly Programming

1 Minimal Syntax

Abstract Type Declaration
Annotation
Annotation Binding
Annotation Type Declaration
Annotation Type Member Declaration
Anonymous Class Declaration
Array Access
Array Creation
Array Initializer
Array Type
Assert Statement
Assignment
Block
Block Comment
Body Declaration
Boolean Literal
Break Statement
Cast Expression
Catch Clause
Character Literal
Child List Property Descriptor
Child Property Descriptor
Class Instance Creation
Comment
Compilation Unit
Conditional Expression
Constructor Invocation
Continue Statement
Default Binding Resolver
Default Comment Mapper
Default Value Pair Binding
Do Statement
Doc Comment Parser
Empty Statement
Enhanced For Statement
Enum Constant Declaration
Enum Declaration
Expression
Expression Statement
Field Access
Field Declaration
For Statement
If Statement
Import Declaration
Infix Expression
Initializer
Instanceof Expression
Javadoc
Labeled Statement
Line Comment
Marker Annotation
Member Ref
Member Value Pair
Member Value Pair Binding
Message
Method Binding
Method Declaration
Method Invocation
Method Ref
Method Ref Parameter
Modifier
Name
Node Event Handler
Node Searcher
Normal Annotation
Null Literal
Number Literal
Package Binding
Package Declaration
Parameterized Type
Parenthesized Expression
Postfix Expression
Prefix Expression
Primitive Type
Qualified Name
Qualified Type
Recovered Type Binding
Recovered Variable Binding
Return Statement
Simple Name
Simple Property Descriptor
Simple Type
Single Member Annotation
Single Variable Declaration
Statement
String Literal
Structural Property Descriptor
Super Constructor Invocation
Super Field Access
Super Method Invocation
Switch Case
Switch Statement
Synchronized Statement
Tag Element
Text Element
This Expression
Throw Statement
Try Statement
Type
Type Binding
Type Declaration
Type Declaration Statement
Type Literal
Type Parameter
Variable Binding
Variable Declaration
Variable Declaration Expression
Variable Declaration Fragment
Variable Declaration Statement
While Statement
Wildcard Type



Atom

List

Method
Pragma
Sequence
Message
Cascade

Block
Return
Variable
Assignment
Literal

3

Reflective Facilities

Reflection

Untercool

Code Generation

(Meta-Programming)

known at compile-time

```
aString asRegex
```

```
IRBuilder new
    pushLiteral: aString;
    send: #asRegex
```

```
Parser parseExpression:
    aString storeString , ' asRegex'
```

```
MessageNode
    receiver: (LiteralNode value: aString)
    selector: #asRegex
```

LISPer know this for a long time

```
``(`,(aString) asRegex)
```

Quoting

Unquoting

scg.unibe.ch/research/helvetia