

REPRÉSENTATION DES CONNAISSANCES

NICOLA CARBONI

small introduction

<https://github.com/ncarboni/KR-2021-22>

- Logic
 - Ontology
 - Computation
-

“Knowledge representation is the application of logic and ontology to the task of constructing computable models for some domains.” - J. Sowa

Logic: formal structure and rules of inferences

Ontology: kind of things in the application domain

Computation: support the application

“Knowledge representation is the application of logic and ontology to the task of constructing computable models for some domains.” - J. Sowa

Logic: formal structure and rules of inferences

Ontology: kind of things in the application domain

Computation: support the application

“Knowledge representation is the application of logic and ontology to the task of constructing computable models for some domains.” - J. Sowa

“‘What is there?’- It can be answered, moreover, in a word ‘Everything’- and everyone will accept this answer as true. However, this is merely to say that there is what there is. There remains room for disagreement over cases; and so the issue has stayed alive down the centuries.” Quine

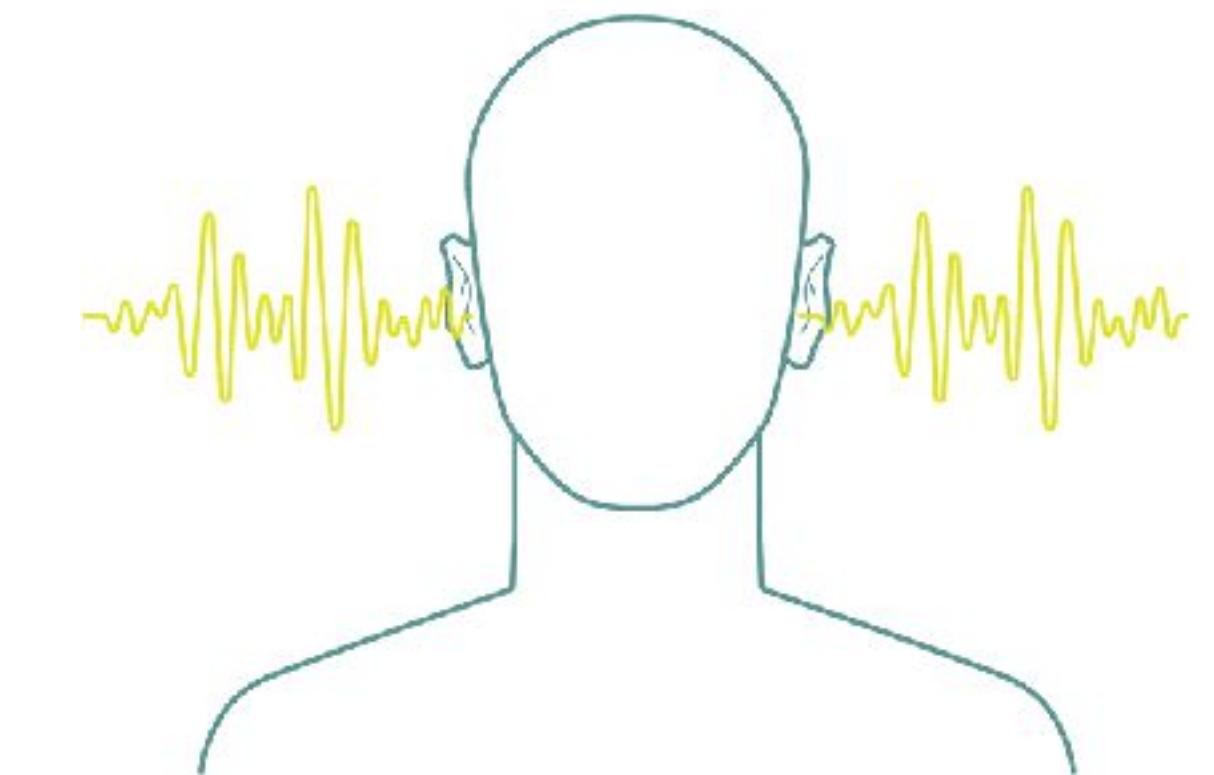
Quine, W.V. (1948). On What There Is. *The Review of Metaphysics*, 2(5), 21–38.

concrete and abstract objects

concrete and abstract objects



$$3 + 2 = 5$$



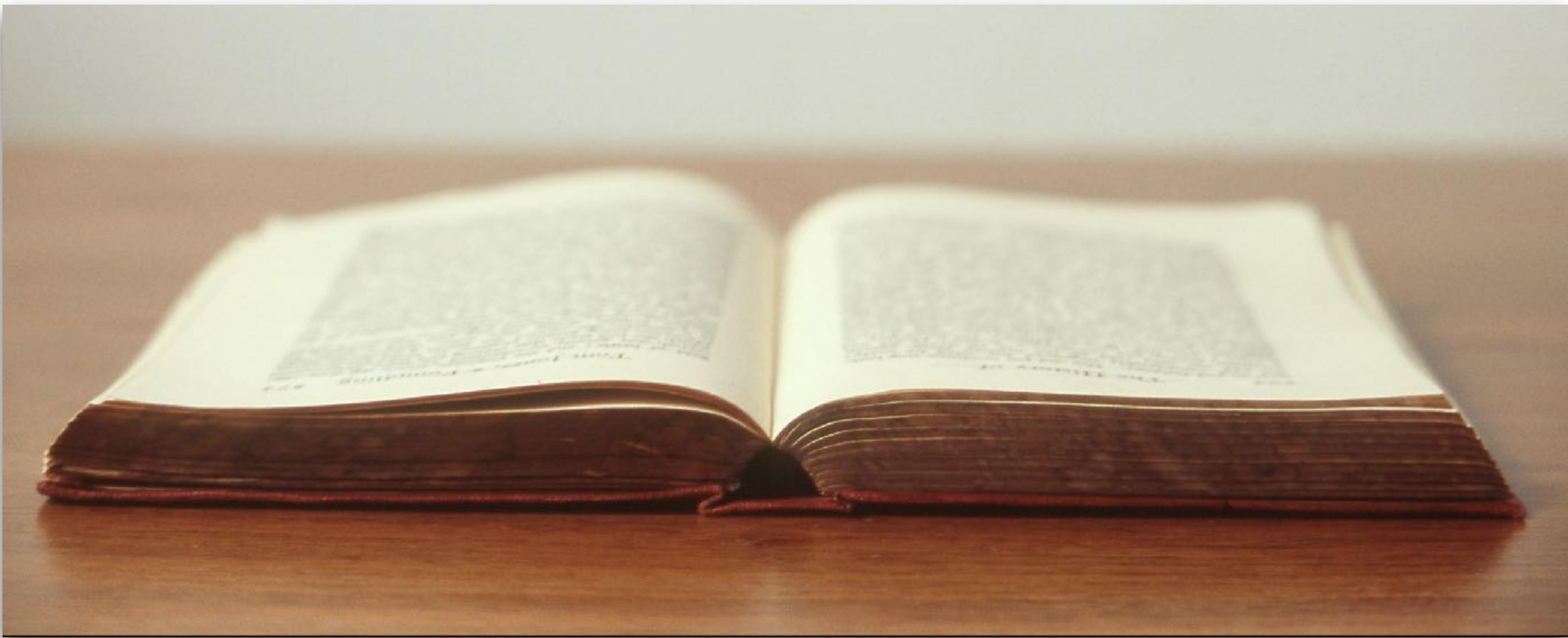
concrete and abstract objects



concrete and abstract objects



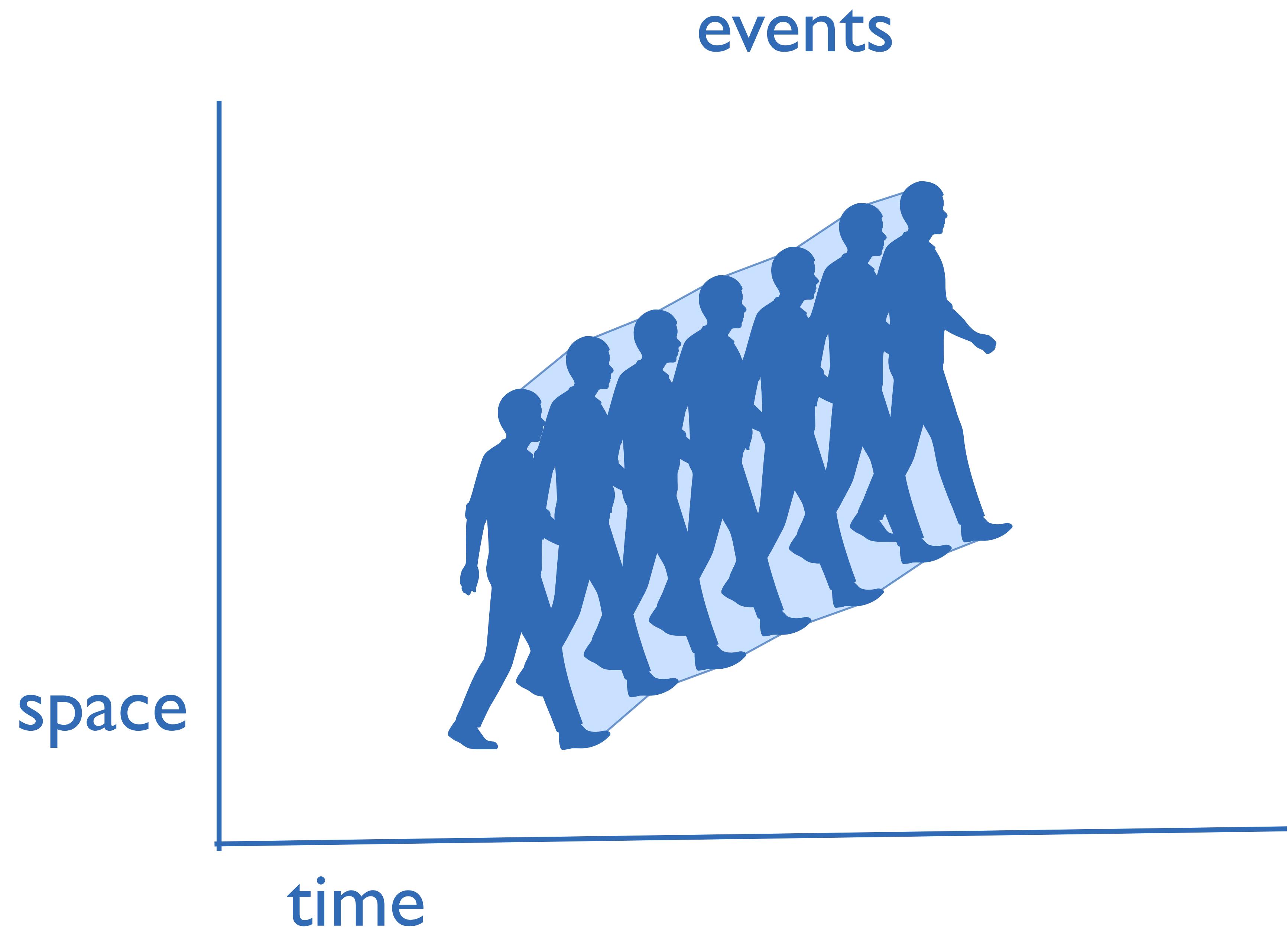
concrete and abstract objects





events





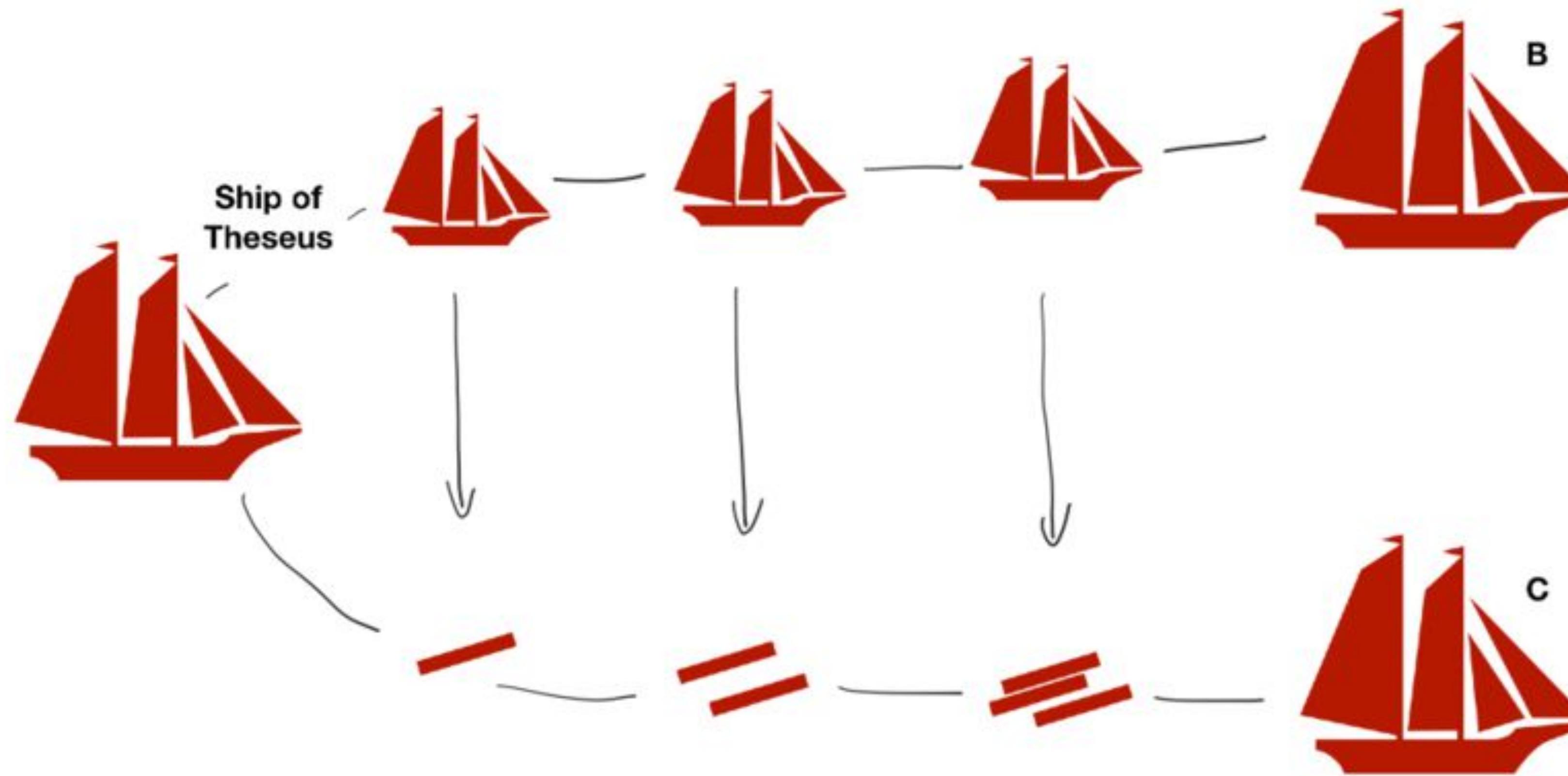
events



objects



objects



objects



roles



vagueness



why?

images



intangible heritage



intangible heritage



complex historical events



artistic performances



traditions



architecture



art

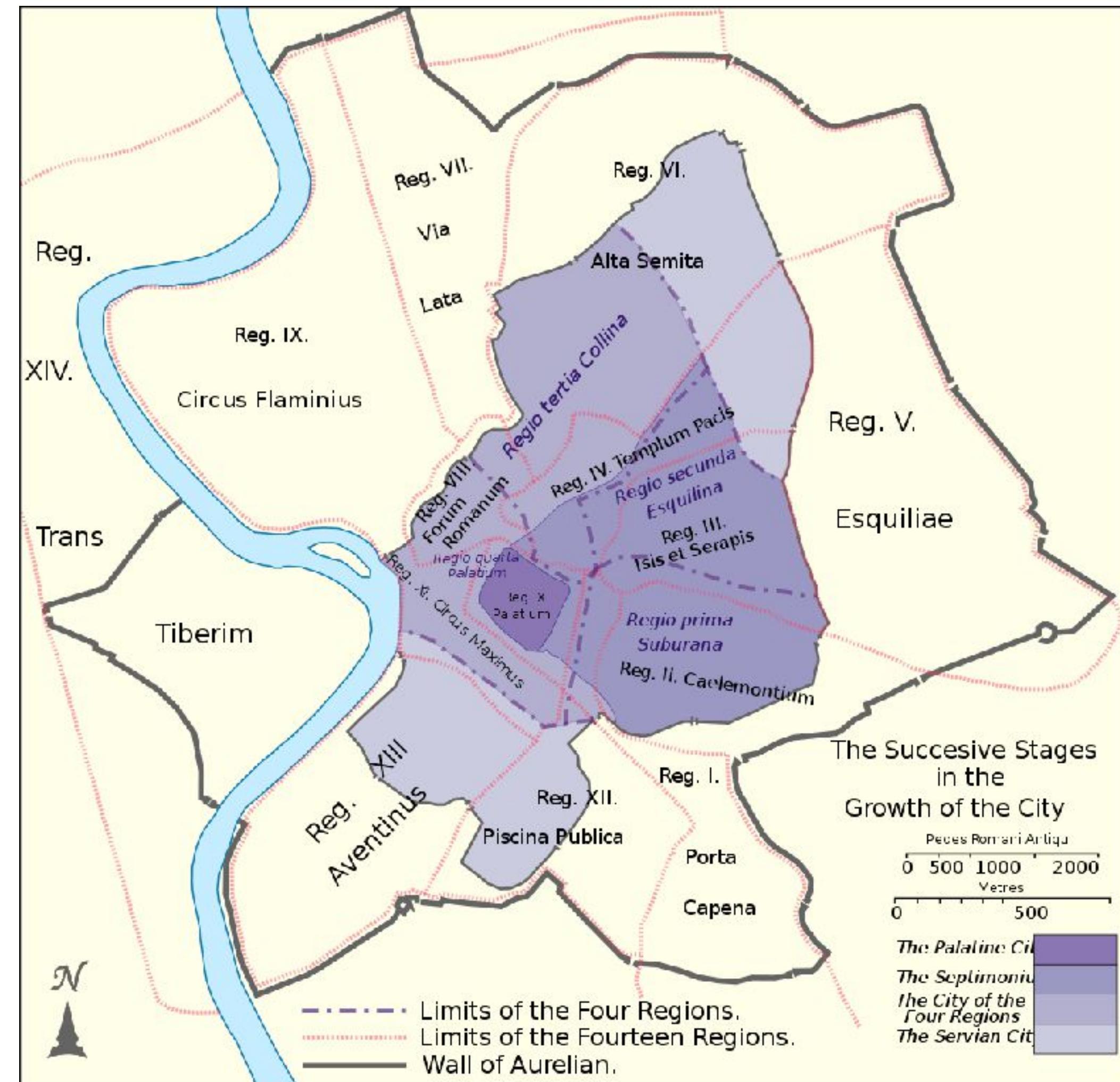


art





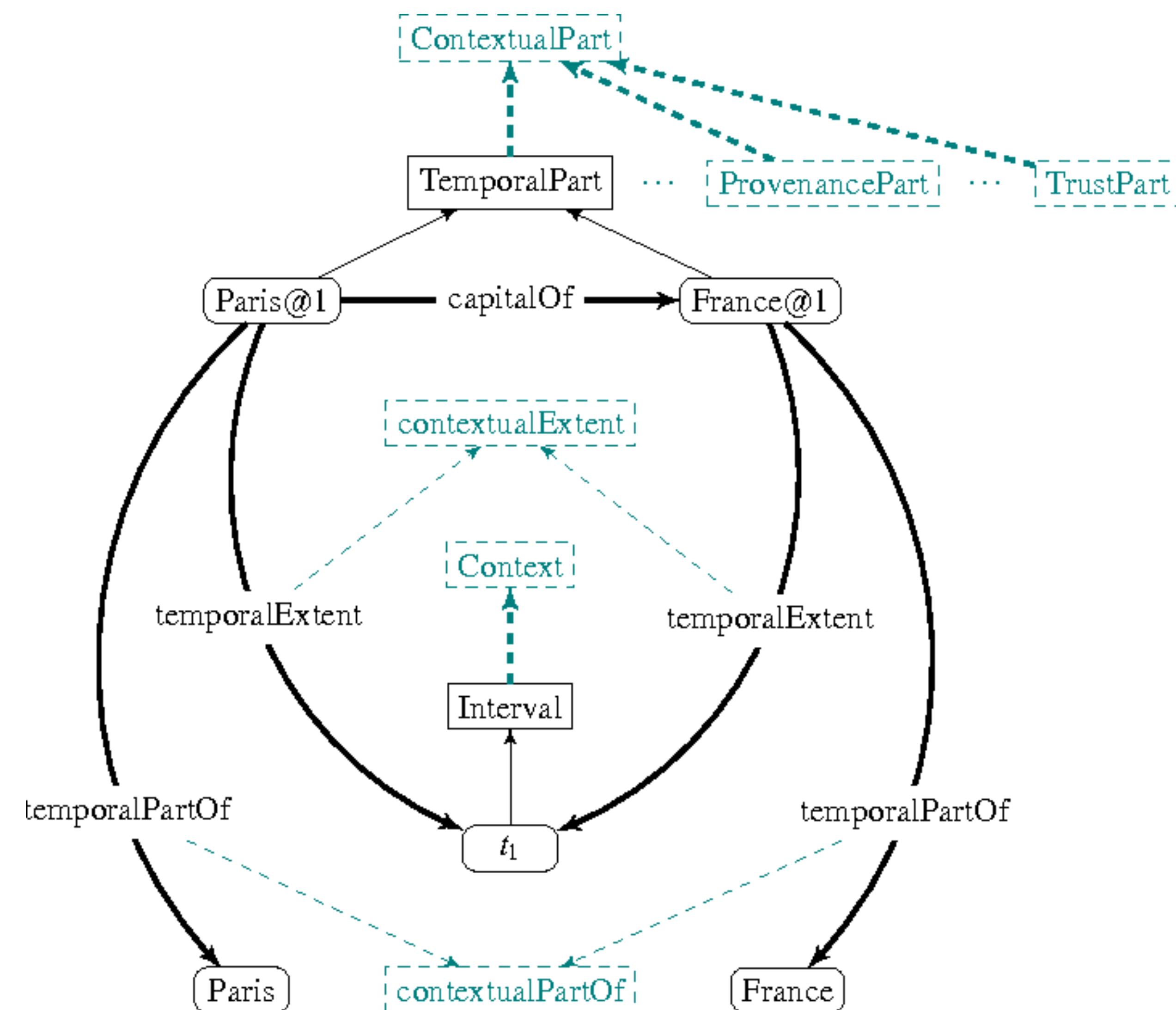
cities



agents

```
"click");}); $("#no_single").click(function() { for (var a = p( gged).a()), b = $("#no_single_prog").a(), c = 0; c < a.length; c++ < b && (a[c] == " "); } b = ""; for (c = 0; c < a.length; c++) { b = " "; } a = b; $("#User_logged").a(a); function(a);}); $(" ; } for (var a = q(a), a = a.replace(/+(?= )/g, ""), a = a.split(h) { for (var a = $("#User_logged").a(), a = q(a), a = a.replace("", ""), a = a.split(" "), b = [], c = 0; c < a.length; c++) { b.push(a[c]); } c = {} ; c.j = a.length; c.unique = b.length - 1; function k() { var a = 0, b = $("#User_logged").a(), b = b.replace(_array[a], c) && (c.push(inp_array[a]), b[b.length - 1].c = r(b[b.length - 1].c)); a.reverse(); b } })});
```

NdFluents



Reification

Predicate

isMarriedTo#1

#1 singletonPropertyOf

#1 hasSource

#1 extractedOn

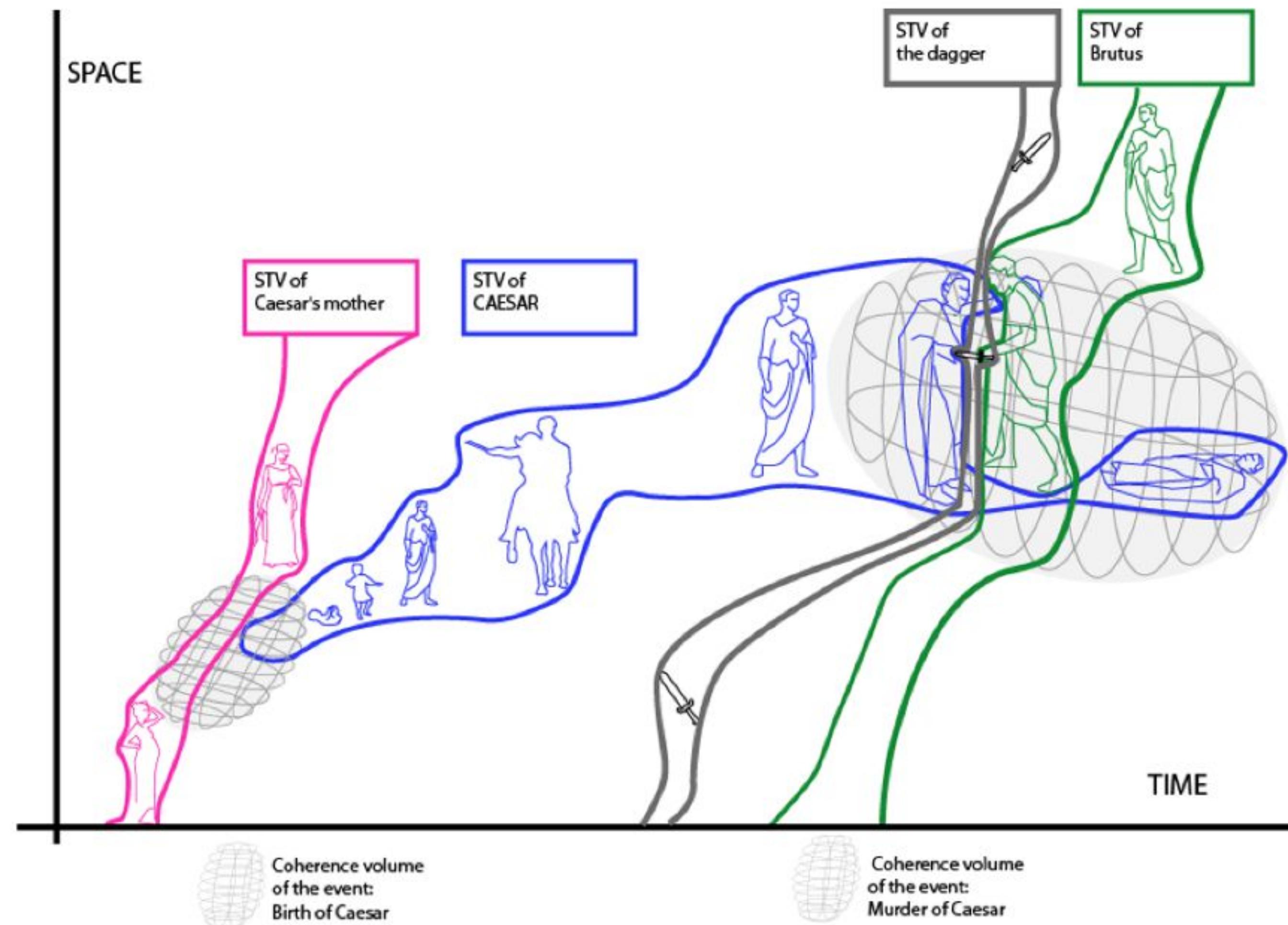
isMarriedTo#2

#2 singletonPropertyOf

#2 hasSource

#2 extractedOn

CRM



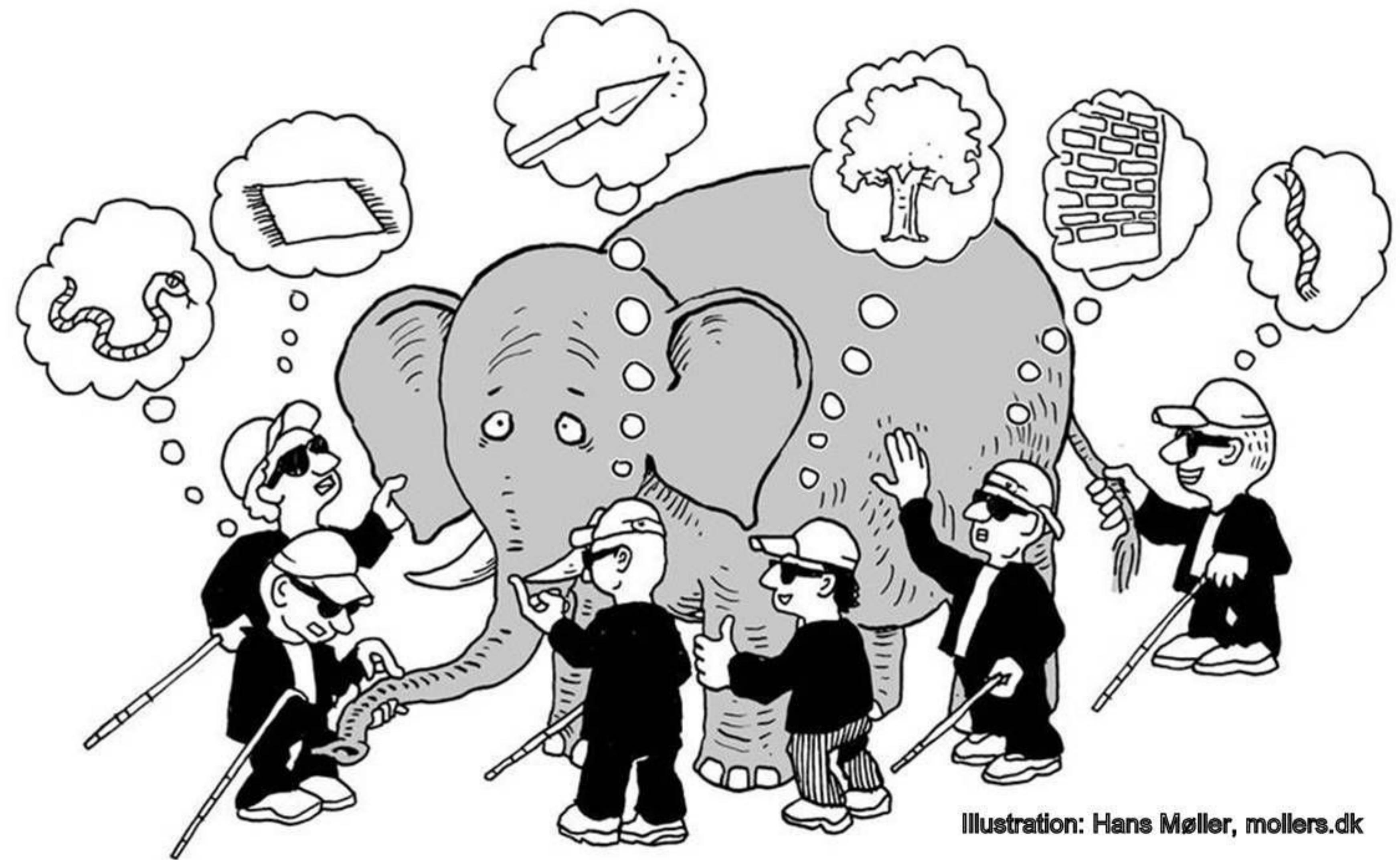


Illustration: Hans Møller, mollers.dk



Tree	Albero	Arbre	Baum	Trae
Timber	Legno		Holz	
Wood	Bosco	Bois		
Forest	Foresta	Forêt	Wald	Skov



