What if you could travel through time
and become someone else?

BACK TO THE MOMENT

Lars Rönnbäck

co-author of Anchor Modeling,
an award winning agile modeling technique
for evolving data environments

@anchormodeling

up to change

www.uptochange.com and www.anchormodeling.com

ANCHOR™

anytime to the time of any

What is it all about?

or how to understand the temporal nature of information

? up to change

It is about liberty

- The Anchor Modeling technique is free to use for anyone in any way you like.
- All material we publish or present is made available through Creative Commons or Copyleft licenses.
- Our modeling tool is Open Source and available in the cloud as well as a stand alone package for local installations.

? up to change

It is about things

- *Things* are that which the domain can recognize and determine if it is something new or something already known. It may be the case that “Leonardo” and “Da Vinci” are the same thing, or it may not. Only the domain knows.

? up to change

It is about identities

- Every thing is given a unique *identity*, through which it is possible to determine equality. If “Leonardo” and “Da Vinci” are referring to the same thing, then “Leonardo” and “Da Vinci” share the same identity.

? up to change

It is about what is said of things

- *Posits* are identifiable statements that take the forms:

x has the property P(t) with value V(t)
where ***x*** is identifiable by the domain, ***P(t)*** is a property that ***x*** has at the time ***t***, and ***V*** is the value that ***P*** takes at the time ***t***.

? up to change

Example posits

- “***Leonardo’s hair color turned gray in 1501”.***
  x = “Leonardo”, which can be mapped to some identity,
  P(t) = hair color, ***V(t)*** = “gray”, ***t*** = 1501

- ***“Leonardo was born in 1452”.***
  x = “Leonardo”, which can be mapped to some identity,
  P(t) = year of birth, ***V(t)*** = “1452”, ***t*** = <u>always</u>

? up to change

… and their relationships

- ***x is related to y_i, i = 1, …, n under the conditions $C_j(t)$ with values $V_j(t)$, j = 1, …, m, defined by the roles R and S_i***

where x and y_i are identifiable by the domain, C_j are optional conditions under which the relationship holds and V_j their values at t, R and S_i are the roles x and y_i have in the relationship.

? up to change

Example posits

- ***"Leonardo is married to Beatrice since 1482".***
  x = *"Leonardo"*, ***y*** = *"Beatrice"*,
  R = husband, ***S*** = wife,
  C(t) = marital status, ***V(t)*** = "married", ***t*** = 1482

- "***Leonardo divorced Beatrice in 1489".***
  x = *"Leonardo"*, ***y*** = *"Beatrice"*,
  R = spouse, ***S*** = other spouse,
  C(t) = marital status, ***V(t)*** = "divorced", ***t*** = 1489

? up to change

It is about classification

- Every thing belongs to a *class*, where classes create groupings of things that through their properties have some kind of similarity. If Leonardo and Beatrice are persons, they may belong to a **Person** class, defined by “that which is born, may have different hair colors over time and can be married”.

? up to change

It is about evolution

- Before 1998 the **Person** class did not have a property called ‘hair color’, but it was added after that, and ‘marital status’ was introduced in 2005. A model is *non-destructively extended* in order to cope with structural changes, such as new classes, new properties, or new relationships.

Every previous version of the database is available as a subset of the current database.

? up to change

Is it about certainty

- There is *no guesswork* in Anchor Modeling, you only model what you know. Whatever happens in the future will be taken care of in the future.
- *Revisiting* your model should be encouraged and inspiring, not discouraged and disconcerting.

? up to change

It is about who said what when

- If ***q*** is a posit, then both the *positors* ***A*** and ***B*** may have stated that ***q*** is a fact, such that:
  A *uttered* ***q*** in 1999
  B also *uttered* ***q***, but in 2003

The recording of such information is done in an *annex* to the posit.

? up to change

It is about concurrency

- Even if ***A*** and ***B*** disagree, say on a value for a property, such that:
  A uttered “Leonardo’s hair was brown.”
  B uttered “Leonardo’s hair was blonde.”

  both views can be recorded, resulting in a *concurrent-temporal* implementation.

? up to change

It is about reliability

- If both ***A*** and ***B*** uttered ***q***, they may have done so with different degrees of reliability:

A (1999): *I think* Leonardo divorced Beatrice in 1489.

B (2003): *I am absolutely certain that* Leonardo divorced Beatrice in 1489.

? up to change

It is about corrections

- Even if ***B*** was so sure about ***q***, there is no guarantee that ***B*** is correct, and perhaps later it is learnt that:

B (2005): *I was wrong*, Leonardo never divorced Beatrice.

As a consequence, the posit made by ***B*** in 2003 must be considered *unreliable*.

we are almost certain that
Leonardo did not have blonde hair.

? up to change

It is about metadata

- If ***B*** uttered ***q*** in 2003, this information may have been *recorded* elsewhere in 2012 by job number 555 extracting data from system 9. This information becomes *metadata* in the model.

There are now three points in time:
when something *changed*, was *posited*, or *recorded*.

? up to change

It is about completeness

- All in all, that means that in 2012, job number 555 using system 9 recorded that: *'In 2003 **B** said "I am absolutely certain that Leonardo divorced Beatrice in 1489".'*
- We also know that 'marital status' was introduced as a property of the **Person** class in 2005.

? up to change

It is about time traveling

- Questions can be given the answers they would have been given at ***any point in time*** and ***by anyone*** making posits, both with respect to which value was or will be in effect at the time and to what the chosen positor knew at the time.
- Furthermore, it is possible to determine if the question was possible to answer at the time or not, due to the structural evolution of the model or when data was recorded.

? up to change

It is about seizing the past and the future

- In Anchor Modeling, you can see the future based on what the future will be like and see the past as the past was really like.
- You can even see the past through what you know today or an alternative future as it would be under different circumstances.

? up to change

ANCHOR™

anytime to the time of any

So how do you do it?

or how to become a vicarious temporal navigator

? up to change

Do it elementary

- Information has no temporality in itself, but information can be temporally modeled:

  Mona Lisa was painted in 1503 and has been hanging in the Louvre since 1797.

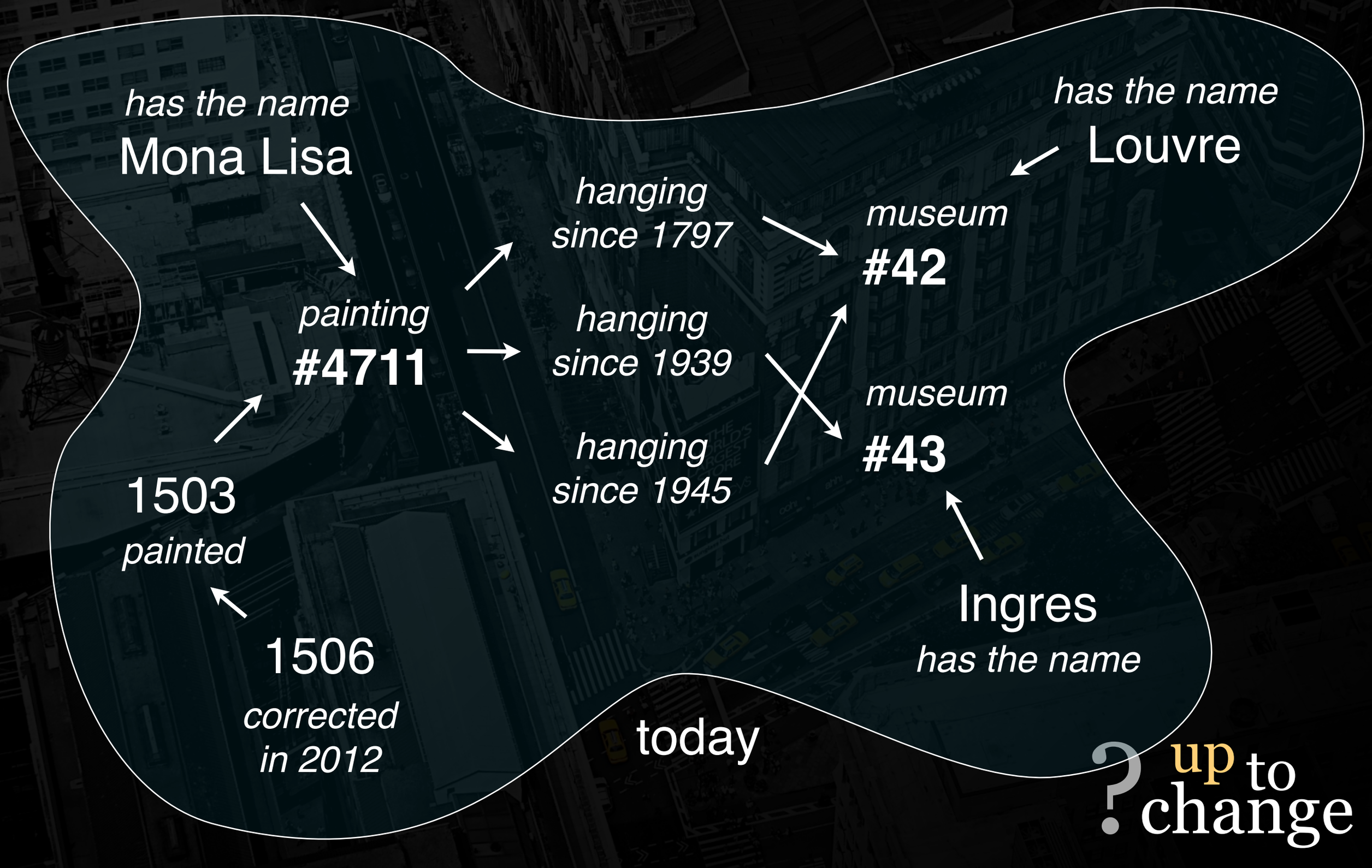
  Research done in 2012 has shown that Mona Lisa was in fact painted somewhat later, in 1506.

  During the Second World War, 1939–1945, the painting was moved to safety in the Ingres Museum.

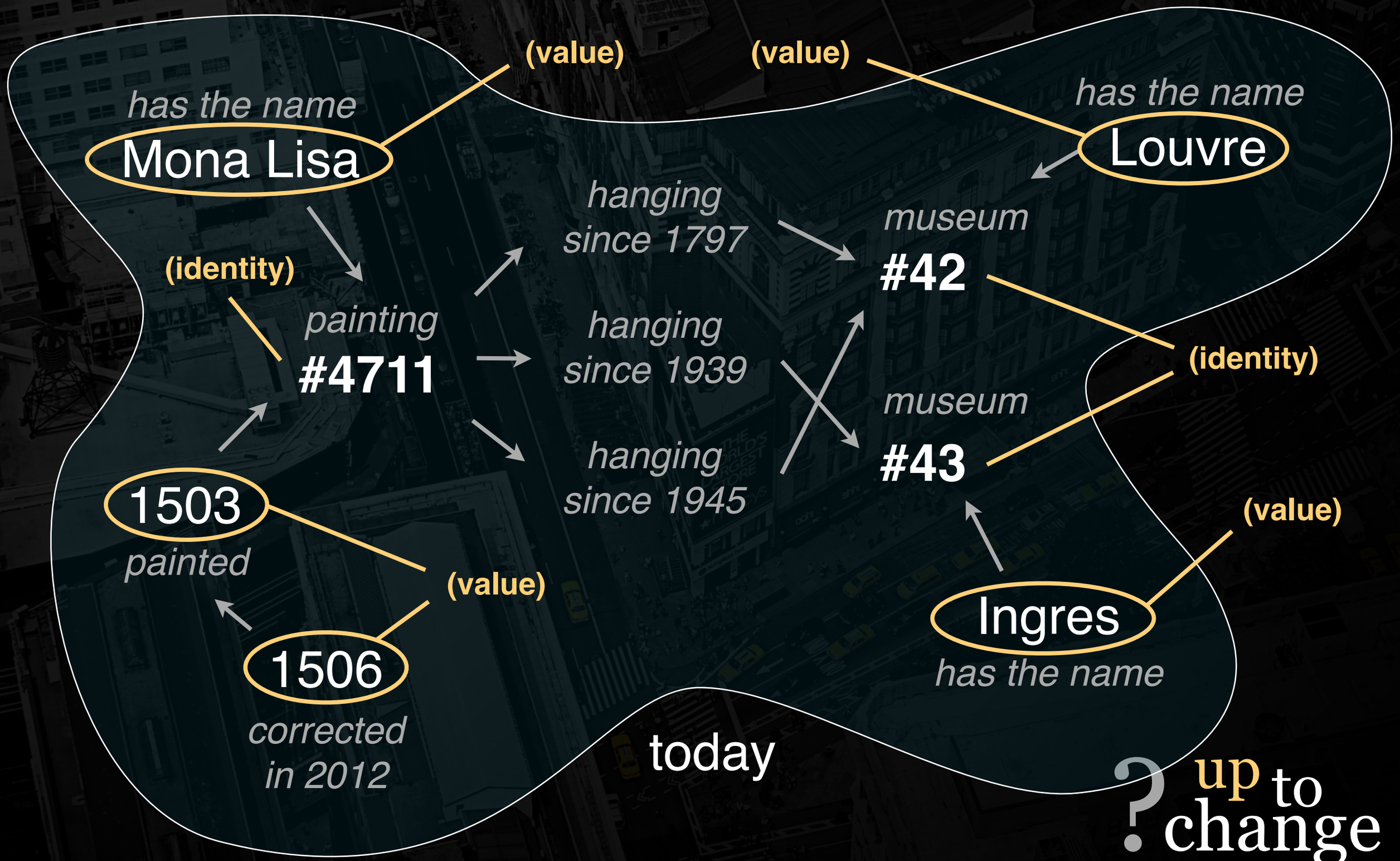
  All this was stored in a database today.

? up to change

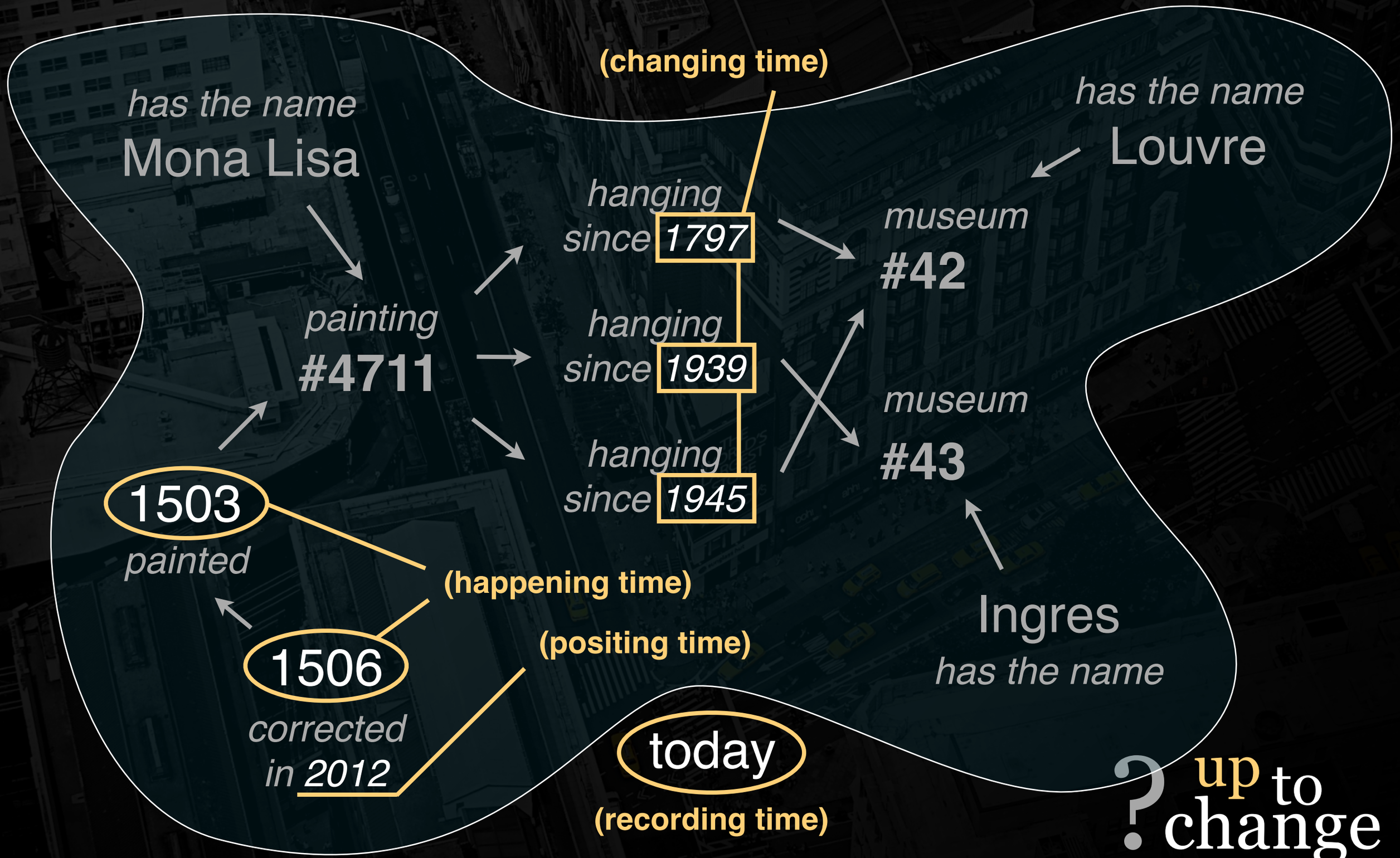
Do it structurally



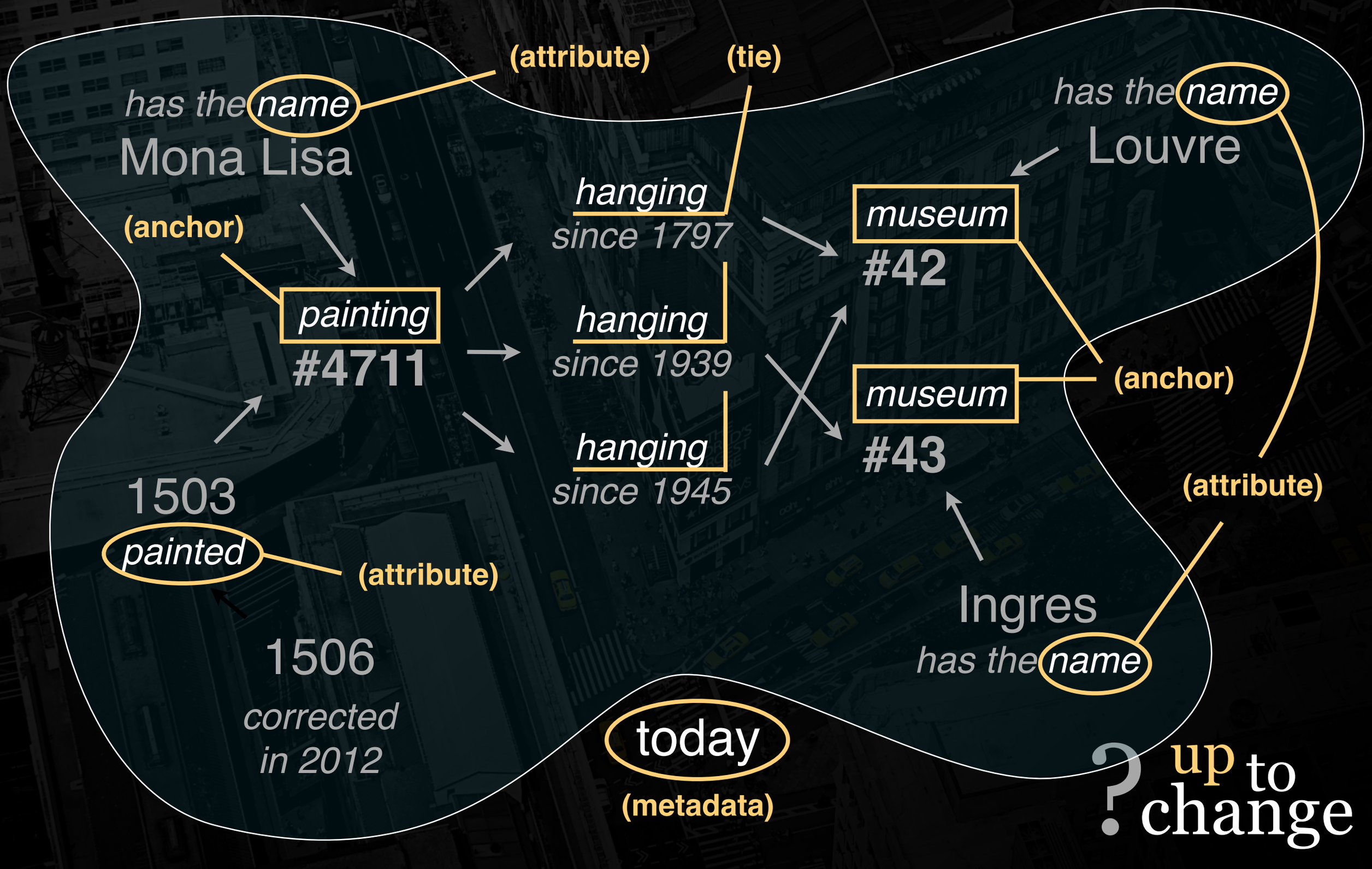
Do it identifiably



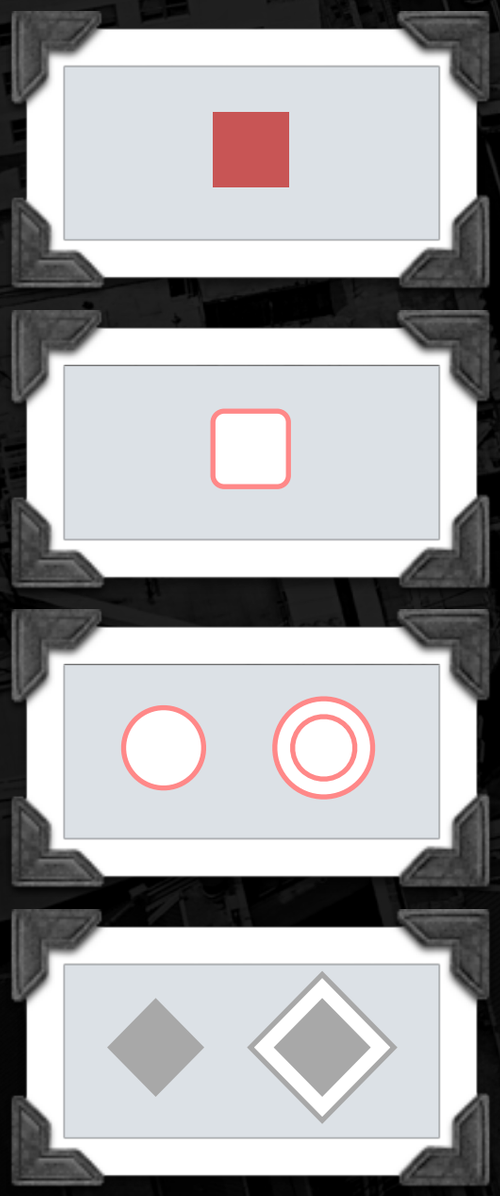
Do it temporally



Do it modeled



Do it conceptually



Anchors
store identities of entities

Knots
store value domains

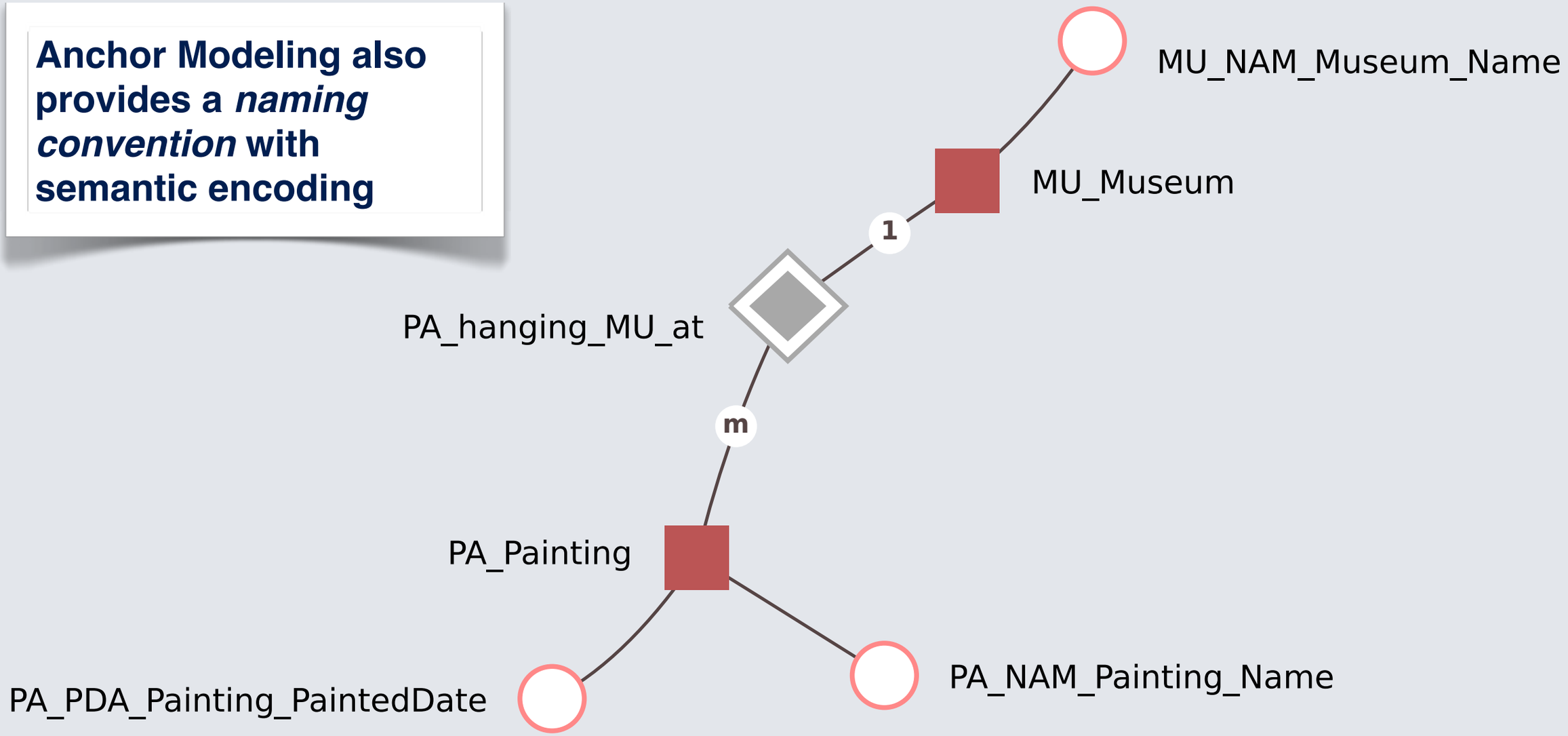
Attributes
store values of properties
(with optional history over changing time)

Ties
store relationships between entities
(with optional history over changing time)

? up to change

Do it graphically

Anchor Modeling also provides a *naming convention* with semantic encoding

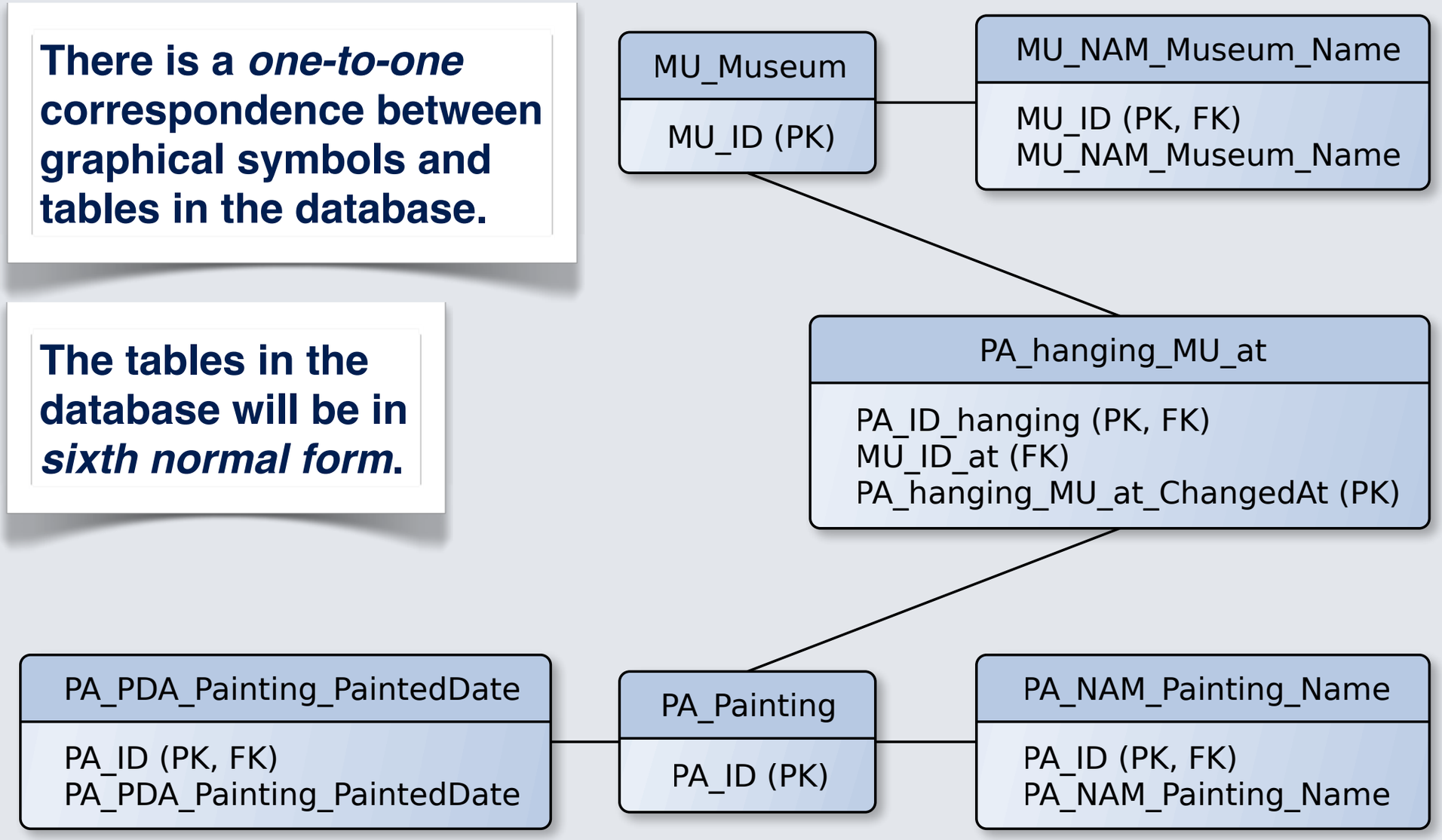


? up to change

Do it relationally

There is a *one-to-one* correspondence between graphical symbols and tables in the database.

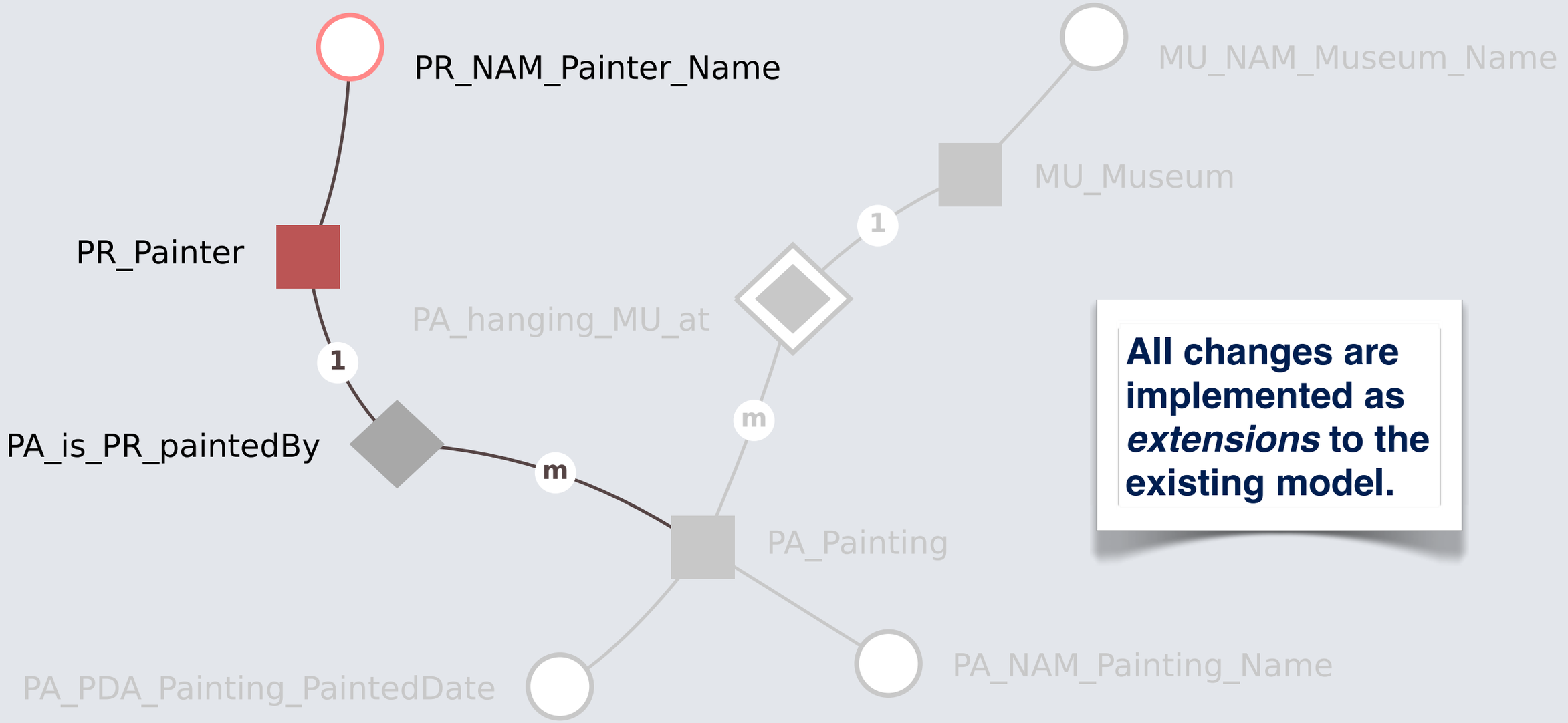
The tables in the database will be in *sixth normal form*.



This is a uni-temporal anchor model.

? up to change

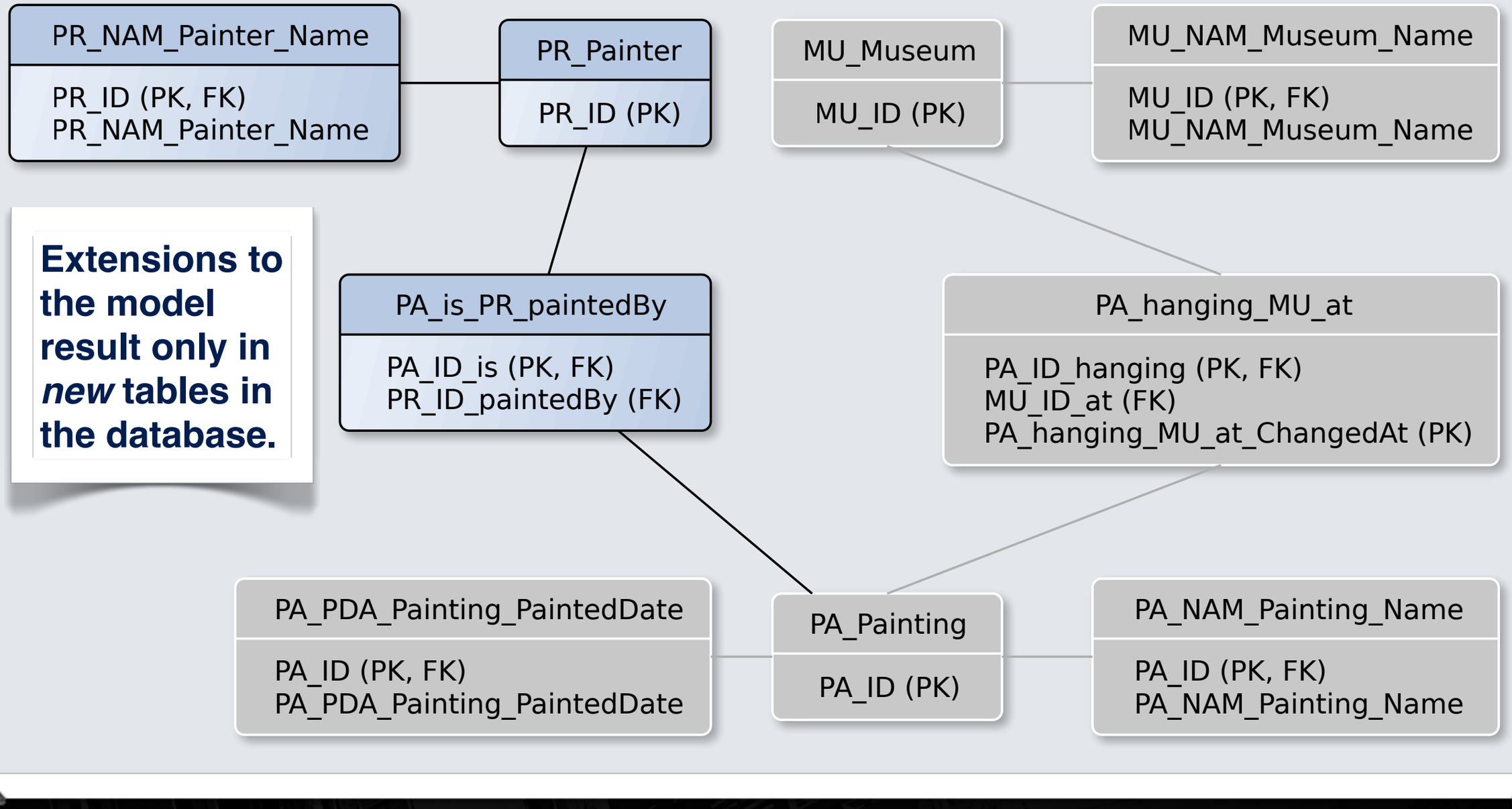
Do it evolutionary



All changes are implemented as *extensions* to the existing model.

? up to change

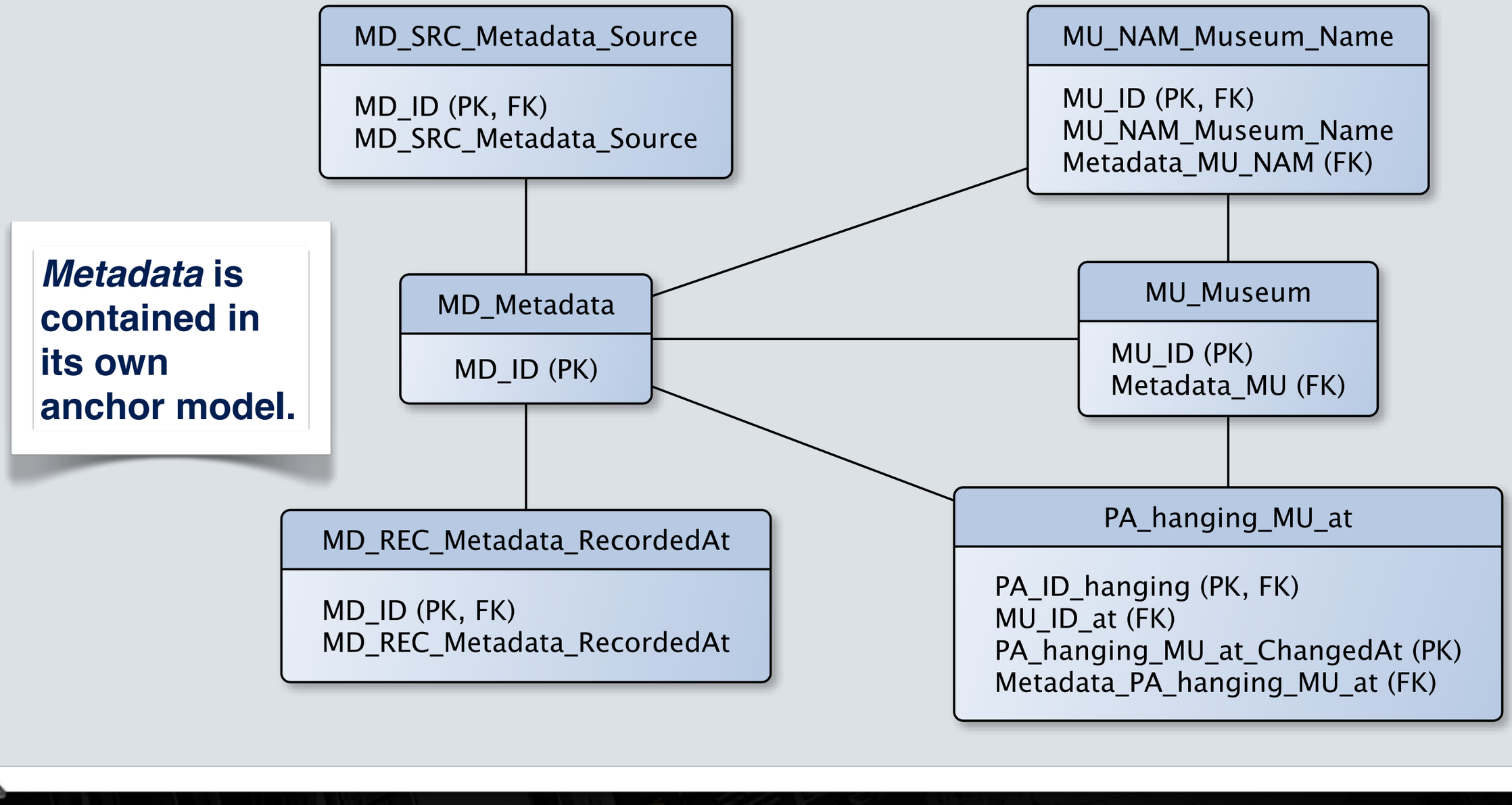
Do it non-destructively



upgrading a database can be done online and almost instantaneously.

? up to change

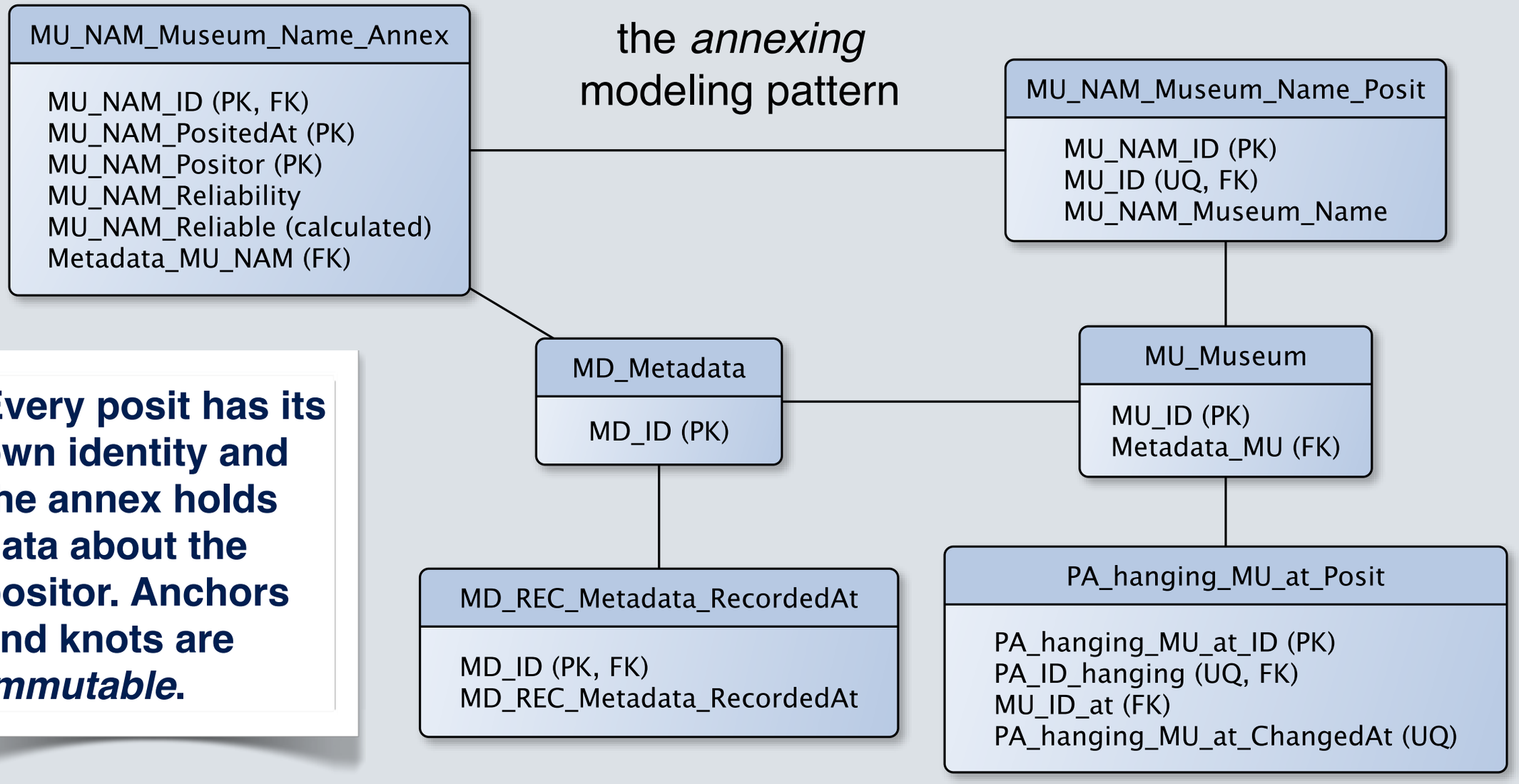
Do it auditable



This is a uni-temporal anchor model with metadata.

? up to change

Do it concurrently



Every posit has its own identity and the annex holds data about the positor. Anchors and knots are *immutable*.

This is a concurrent-temporal anchor model with metadata.

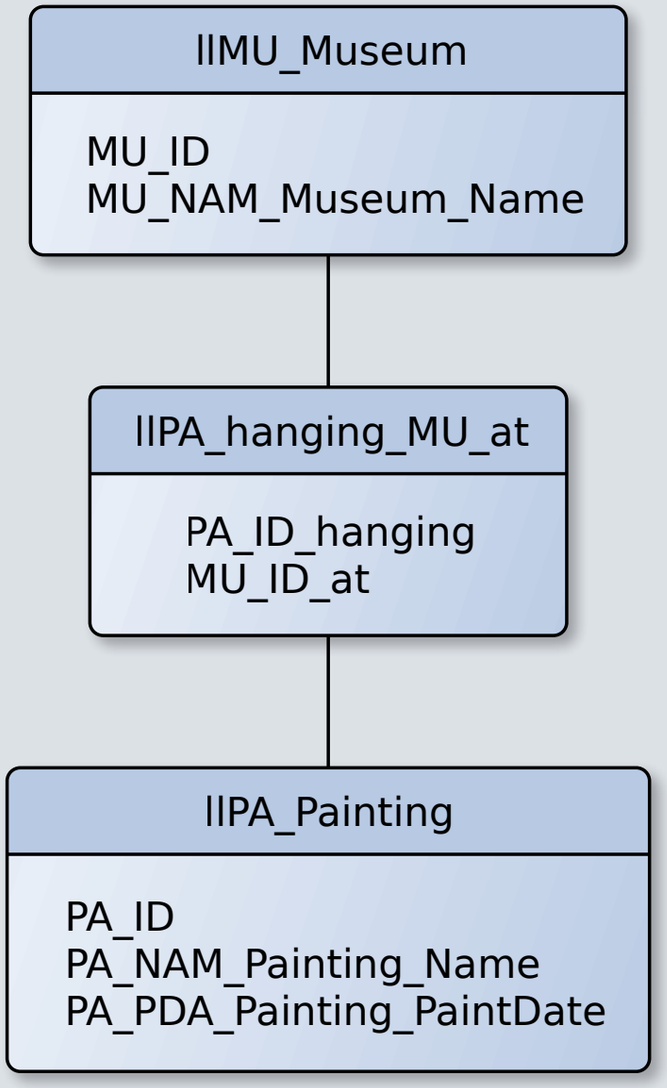
? up to change

Do it with perspective

Most end users are familiar with 3NF and need not see the underlying 6NF model.

Users only need to pick the temporal perspectives suitable for their task.

PERSPECTIVES
current
latest
point-in-time
difference
natural



This is a temporal perspective of an anchor model.

? up to change

Do it table-like

```
-- happening time
insert into llPA_Painting (
  PA_NAM_Painting_Name,
  PA_PDA_Painting_PaintDate
) values ('Mona Lisa', 1503);

-- changing time
update llPA_hanging_MU_at (
set
  MU_ID = 43,
  PA_hanging_MU_at_ChangedAt = 1939
where
  PA_ID = 4711;

-- recording time
delete from llPA_Painting
where
  PA_NAM_Painting_Name = 'Mona Lisa';

insert into llPA_Painting (
  PA_NAM_Painting_Name,
  PA_PDA_Painting_PaintDate
) values ('Mona Lisa', 1506);
```

```
-- latest changing latest positing
select * from llPA_Painting;

-- point-in-changing latest positing
select * from plPA_Painting(1942);

-- point-in-changing point-in-positing
select * from ppPA_Painting(1942, 1942);
```

The latest and complete views behave very much like tables thanks to insert, update, and delete-triggers.

? up to change

ANCHOR™

anytime to the time of any

Who is already doing it?

or how some have learnt to adapt to change

? up to change

The insurance industry is doing it

- A customer lifetime value model is using concurrent-temporal modeling to store the input values for over 600 parameters, each estimated *(changing time)* for the next 50 years, and revised *(positing time)* yearly.
- Such a set is called a scenario *(positor)*, and there are for example standard, best-case, and worst-case scenarios that can be used in the calculations.

? up to change

The clinical domain is doing it

- A hospital is like a fast changing micro-environment *(evolution)*, where you have transactions between clinics, research, biobanks, and laboratories.
- These use Anchor Modeling in order to create a federated database layer of the various models.
- Federation normally means a lot of work is being done at query-time, but in AM it is almost trivial *(non-destructive)*.

? up to change

The traders are doing it

- A trading system is looking back at every second of the last 15 minutes of trading, in order to predict the next 60 seconds *(changing time)* of trades with different probabilities *(reliability)*.
- Statistical and genetic systems analyze the 60x60 predictions per minute over longer periods *(time-traveling)*, and adjust the trading algorithm.

? up to change

The media distributors are doing it

- In a strictly regulated market, a media distributor has built an enterprise data warehouse from which it is possible to create reports based on the exact circumstances that were in effect at the close of each period *(changing time)*.
- It is also possible to trace every single piece of information back to its source *(metadata)*.
- Any historical report can be recreated at any point in time *(time-traveling)*, which meant that reports no longer have to be printed out and kept in binders.

? up to change

ANCHOR™

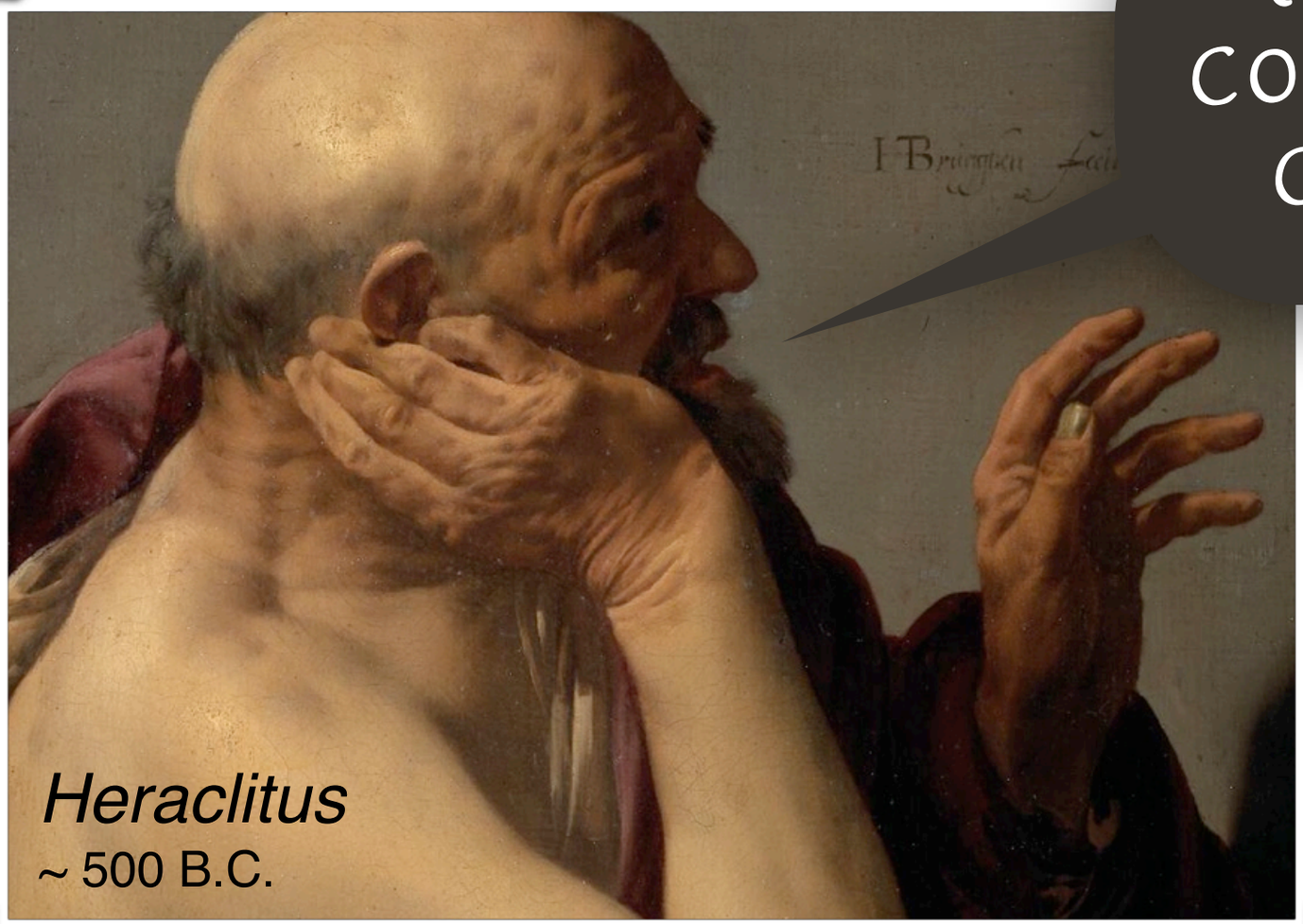
anytime to the time of any

And why should You?

or how to survive in an ever changing environment

? up to change

Because change is everywhere



Heraclitus
~ 500 B.C.

THE ONLY CONSTANT IS CHANGE.

up to change?

Because change is accelerating

5,000,000,000 The earth forms
500,000,000 Vertebrates
50,000,000 Mammals
5,000,000 Primates
500,000 Humans
50,000 Great migrations
5,000 Civilizations
500 Printing press
50 Television
5 Mobile Internet

? up to change

Because only those who adapt survive

- Big Data
- Moore’s Law
- Ephemeralization
- New emerging markets
- Customer “re-preferencing”
- Volatile economic environment
- Fast changing and far reaching regulations

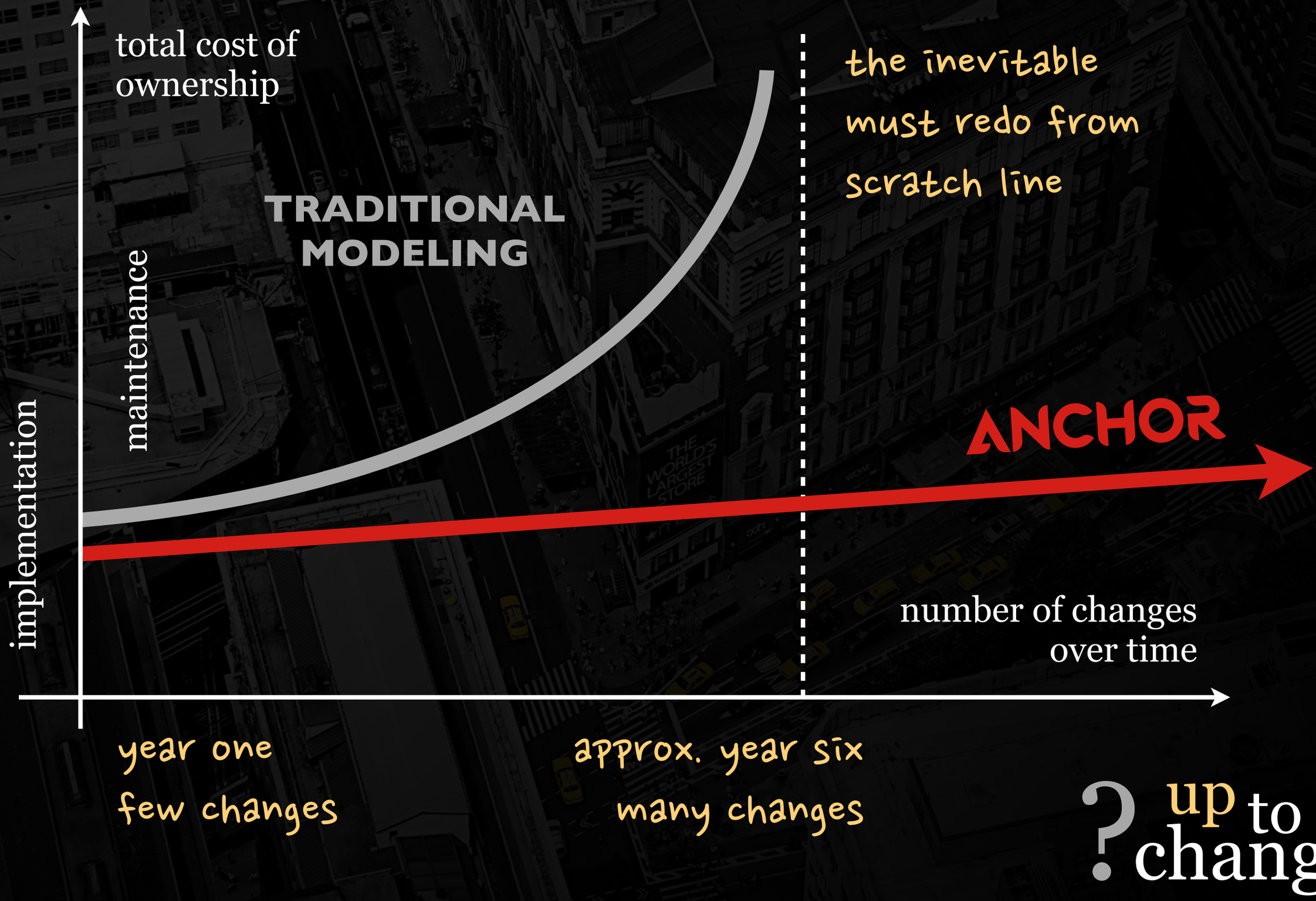
? up to change

Because others get quicker and dirtier

Few things scale well with change

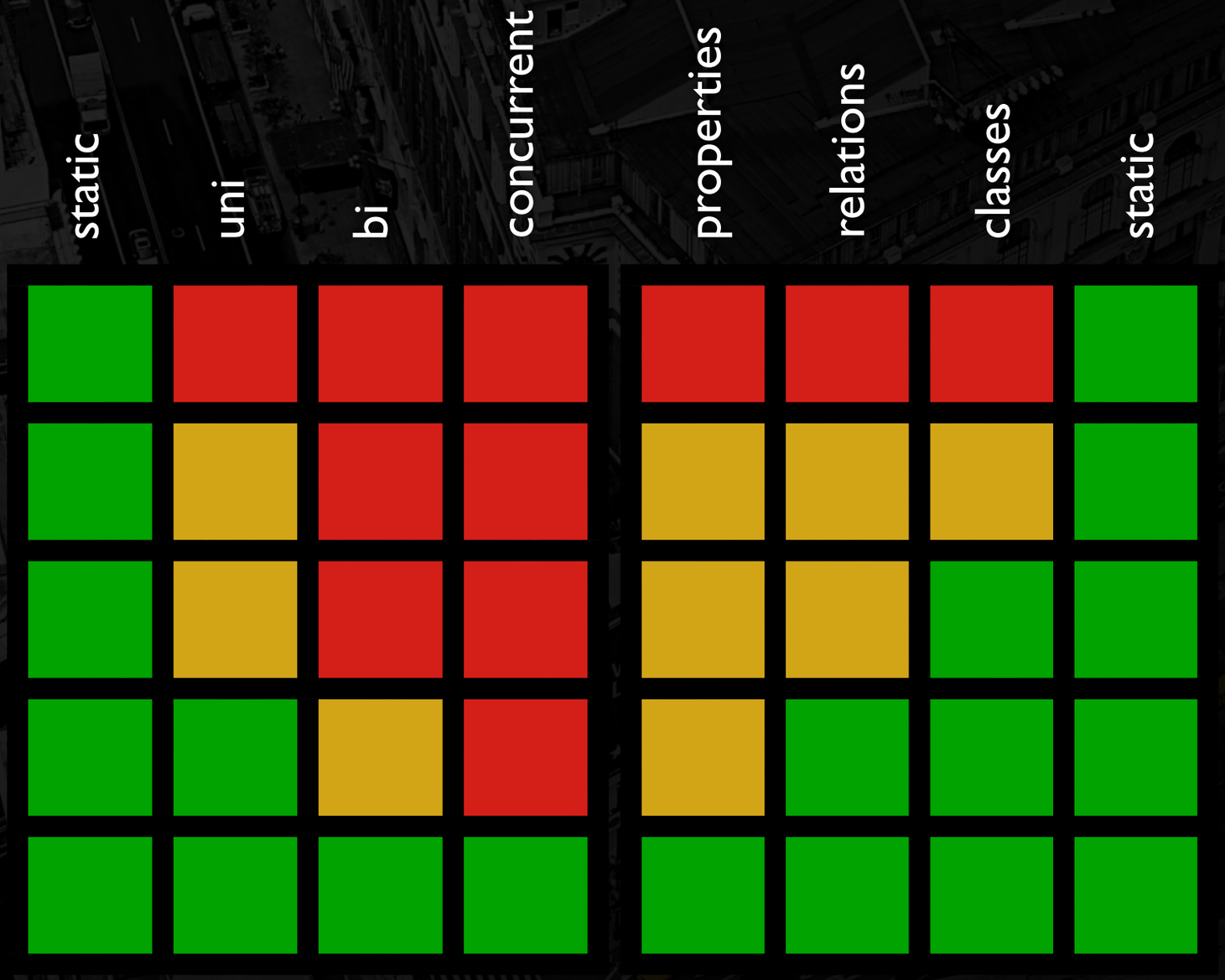
? up to change

Because of money



? up to change

Because of time



? up to change

Because it is well-founded

- Entity Relationship Modeling (1976 – *Chen*)
- Bitemporal Databases (1992 – *Snodgrass*)
- The Sixth Normal Form (2002 – *Date, Darwen, Lorentzos*)
- Immutability, Temporal Independency, and Annexing (2009 – *Rönnbäck, Regardt, Johannesson, Bergholtz, Wohed*)

Anchor Modeling is based on well known research.

? up to change

Because it is proven and active

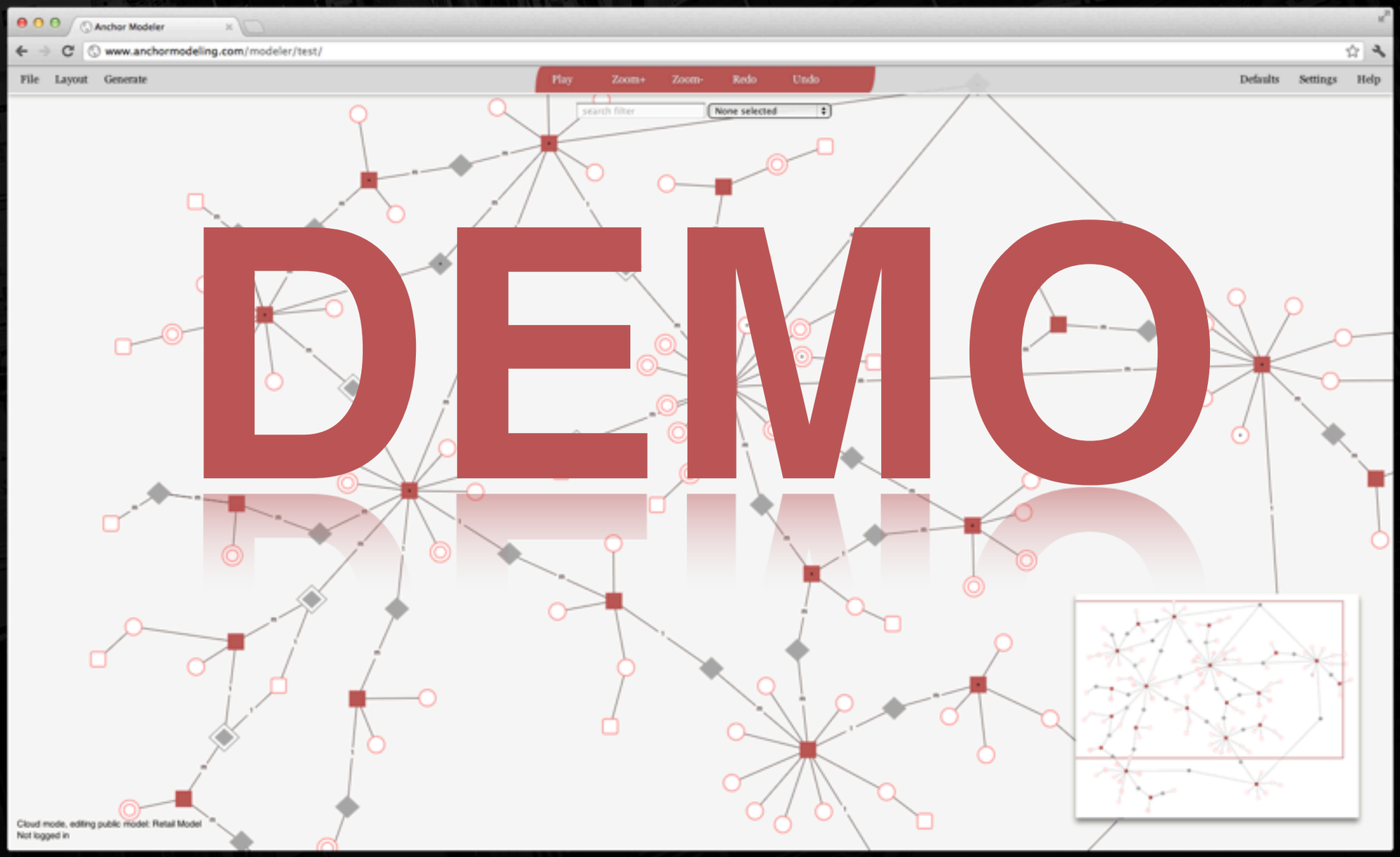
Research Group

Professor Paul Johannesson, DSV
M.Sc. Lars Rönnbäck, Up To Change
M.Sc. Olle Regardt, Teracom
Ph.D. Maria Bergholtz, DSV
Ph.D. Petia Wohed, DSV

Anchor Modeling is developed since 2003 in a collaboration between the Swedish industry and academia, the Department of Computer Science at Stockholm University (DSV).

? up to change

Because of powerful tools



? up to change

Anchor Modeling ...

- has a solid scientific formalization.
- is built on well known principles.
- is easy to learn.
- is hard to make mistakes with.
- fully supports agile development.
- shortens implementation time.
- lowers maintenance costs.
- preserves all previous versions of the database.
- increases the lifetime of the database.
- has Open Source tools.
- is free to use.

GET INVOLVED!

? up to change

ANCHOR™

- Homepage: www.anchormodeling.com
- E-mail: lars.ronnback@anchormodeling.com
  sales@uptochange.com
- Twitter: anchormodeling
- LinkedIn: Anchor Modeling Group
- Facebook: Anchor Modeling
- Wikipedia: Anchor Modeling
- MSDN: Anchor Modeling

? up to change

on a side note:

Temporal Dimensional Modeling

- Using techniques from Anchor Modeling, dimensional modeling can be made uni-temporal without the need for slowly changing dimensions!
- We can even take it one step further, and introduce concurrency in a star schema.
- It can be done without sacrificing performance.
- Scientific paper is going to be published in 2013.

? up to change