

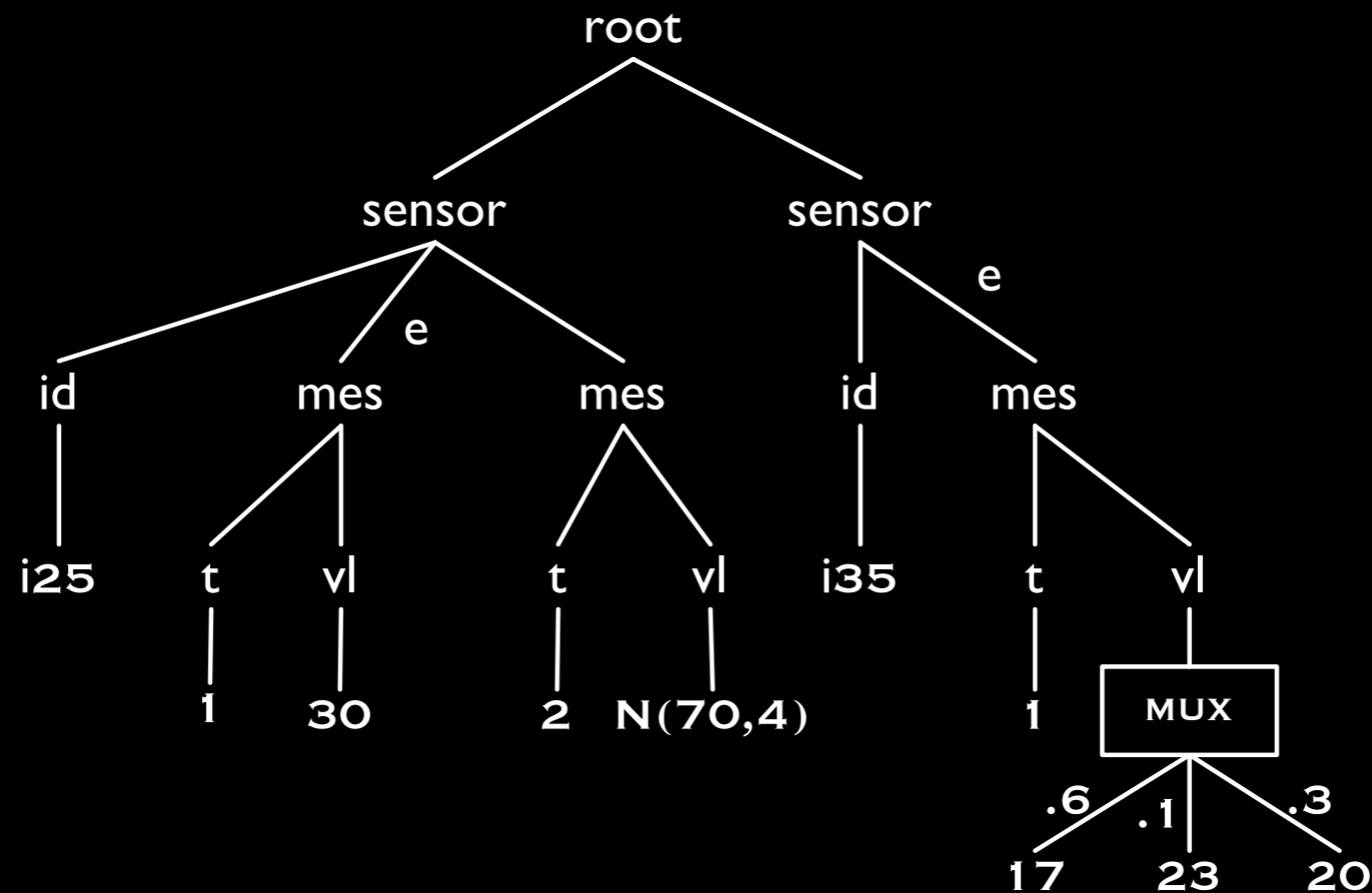
# Modeling and Querying Uncertain Data with Probabilistic XML

S. Abiteboul,<sup>\*</sup> T-H. H. Chan,<sup>+</sup> E. Kharlamov,<sup>\*@</sup> W. Nutt,<sup>@</sup> P. Senellart<sup>#</sup>

<sup>\*</sup>INRIA Saclay Gemo, <sup>+</sup>Max-Planck-Institut für Informatik,  
<sup>@</sup>Free University of Bozen-Bolzano, <sup>#</sup>Télécom ParisTech

DataRing meeting, Nantes, July 2009

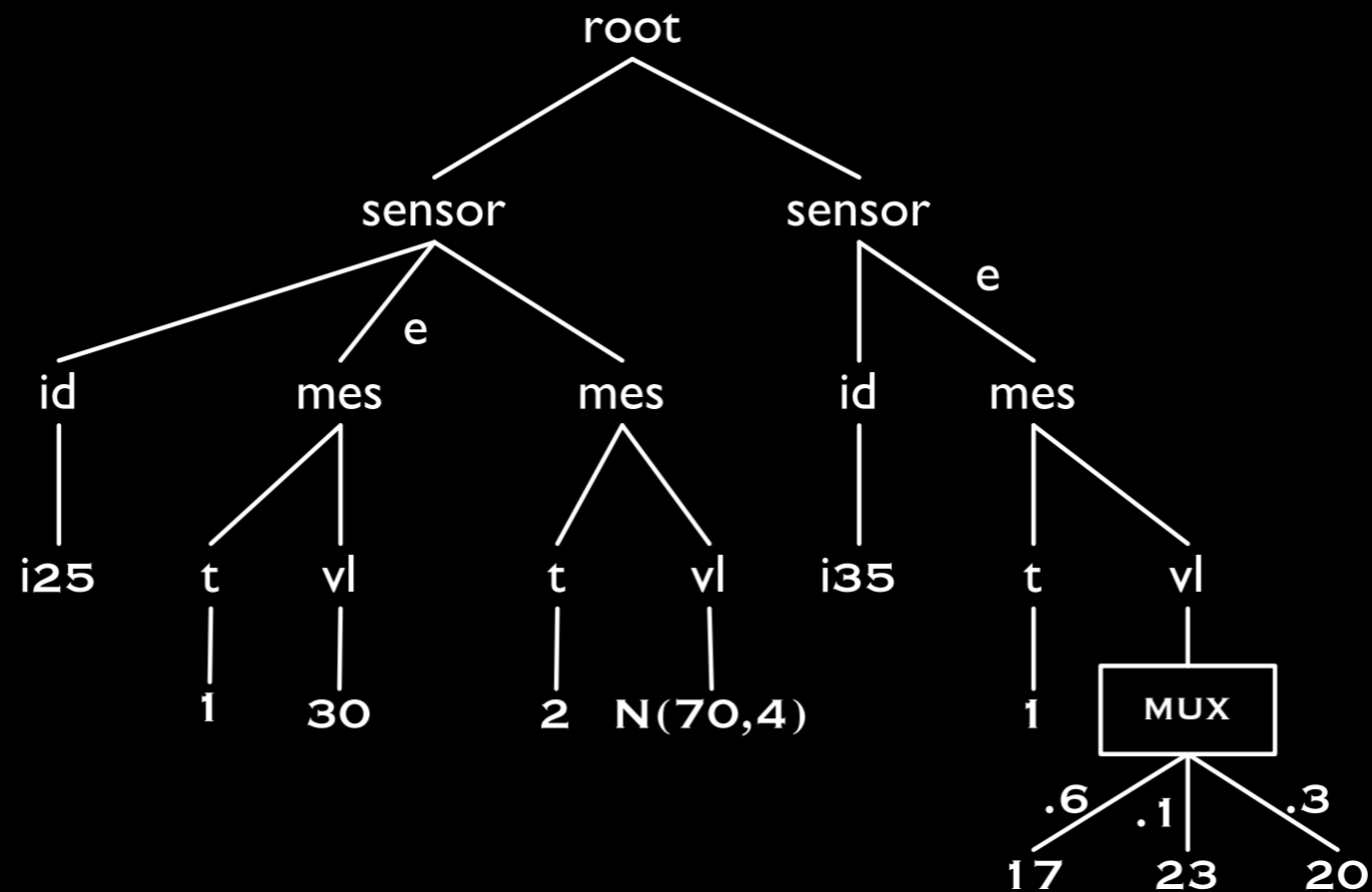
# Scenario: Temperature Monitoring



- mes - measurement
- t - time
- vl - value

- e - event “it does not rain”  
 $\Pr(e) = .4$
- MUX - mutually exclusive options
- $N(70,4)$  - normal distribution

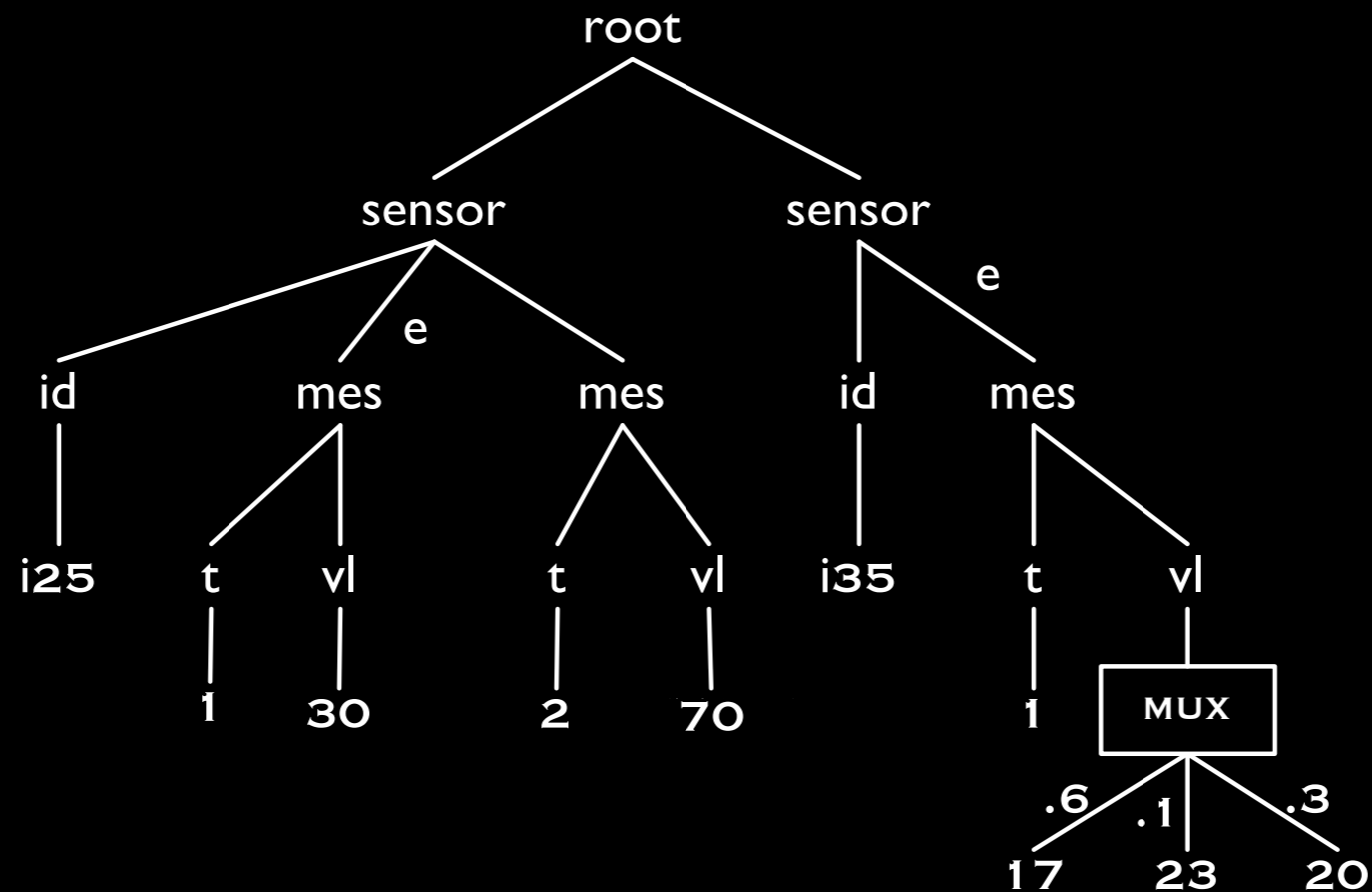
# Simplified Scenario



- mes - measurement
- t - time
- vl - value

- e - event “it does not rain”  
 $\Pr(e) = .4$
- MUX - mutually exclusive options
- $N(70,4)$  - normal distribution

# Simplified Scenario



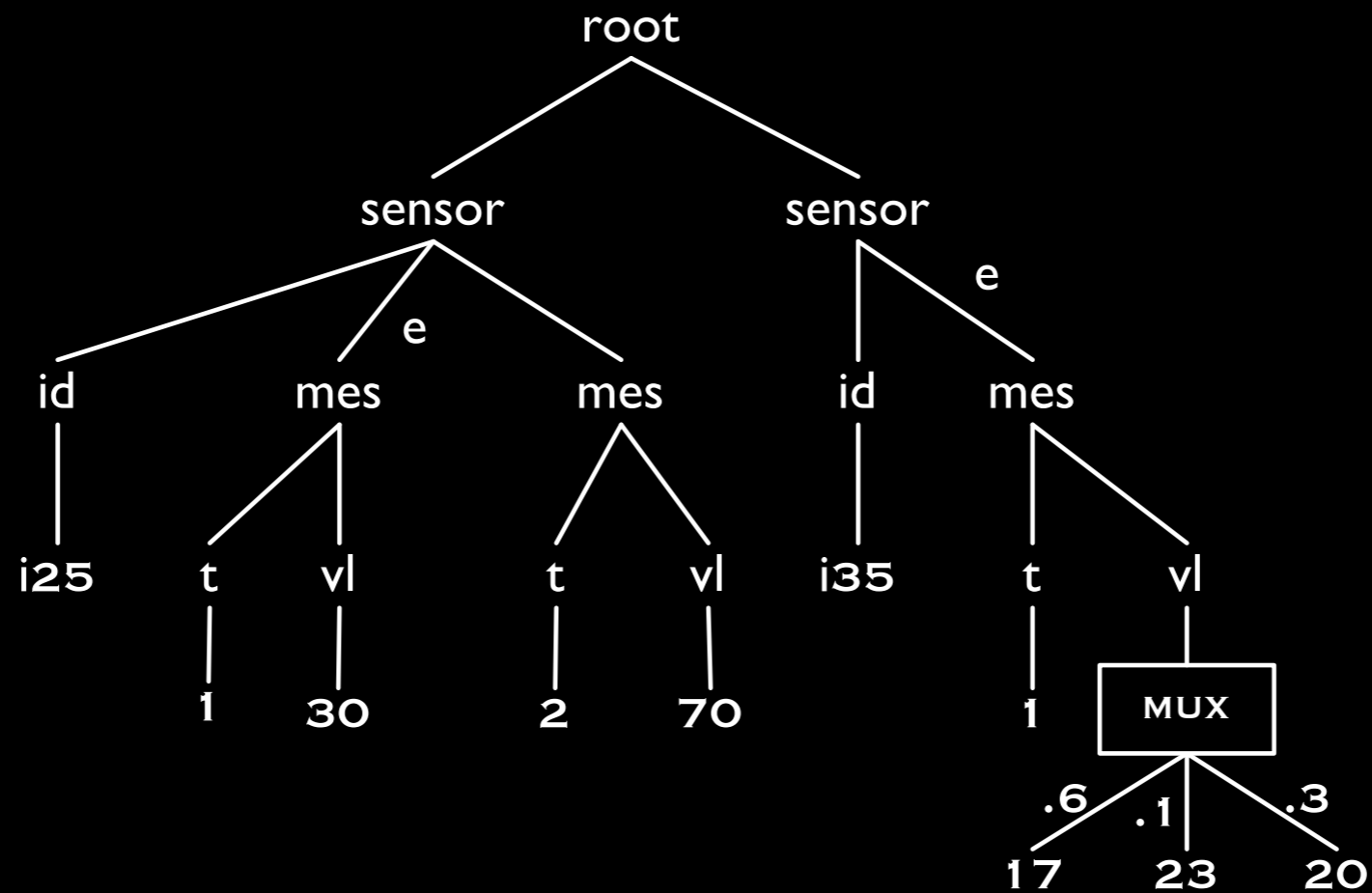
- mes - measurement
- t - time
- vl - value

- e - event “it does not rain”  
 $\Pr(e) = .4$
- MUX - mutually exclusive options

# Probabilistic XML Documents

- Extend XML documents with
  - **Distributional** nodes (e.g. PXML-MUX)
  - **Events** that mark up edges (PXML-Events)
- Describe probability **distributions** over the set of all XML documents

# Possible Worlds Semantics

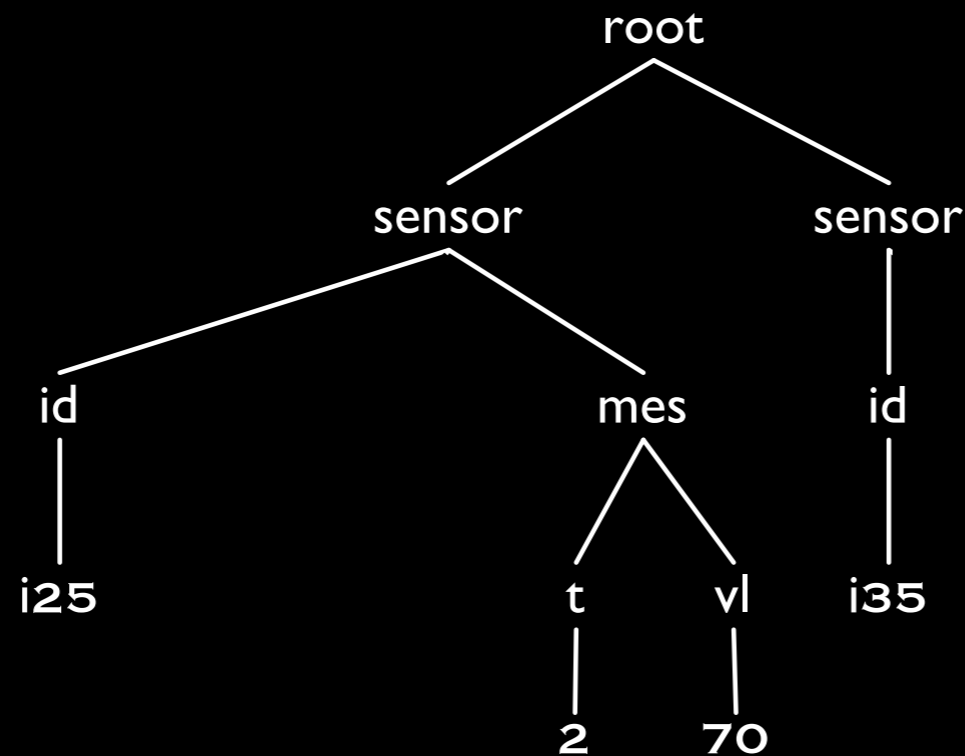


World W:

- $e = \text{false}$  (rain)

$$\Pr(W) = .6$$

# Possible Worlds Semantics

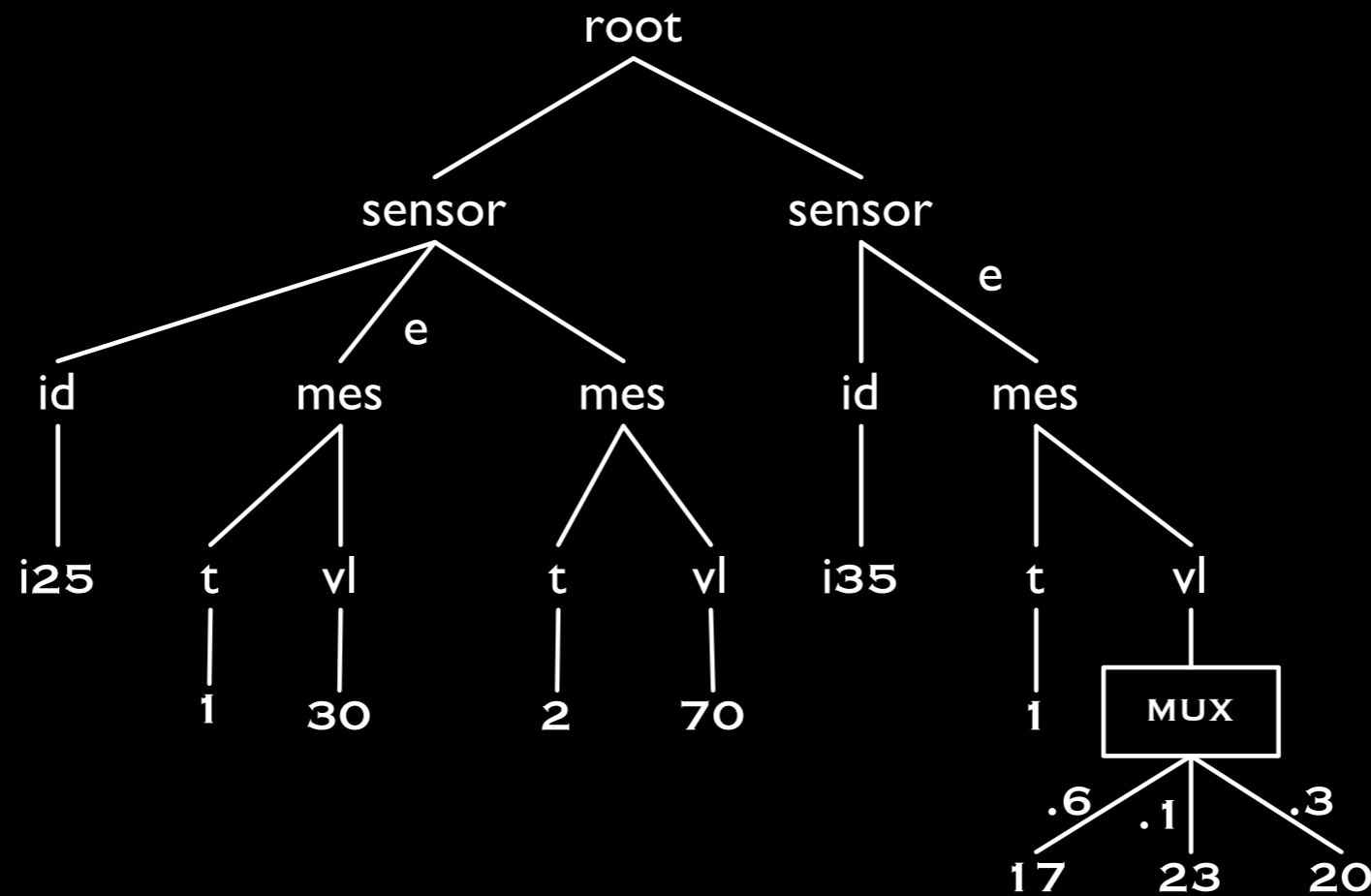


World  $W$ :

- $e = \text{false}$  (rain)

$$\Pr(W) = .6$$

# Possible Worlds Semantics

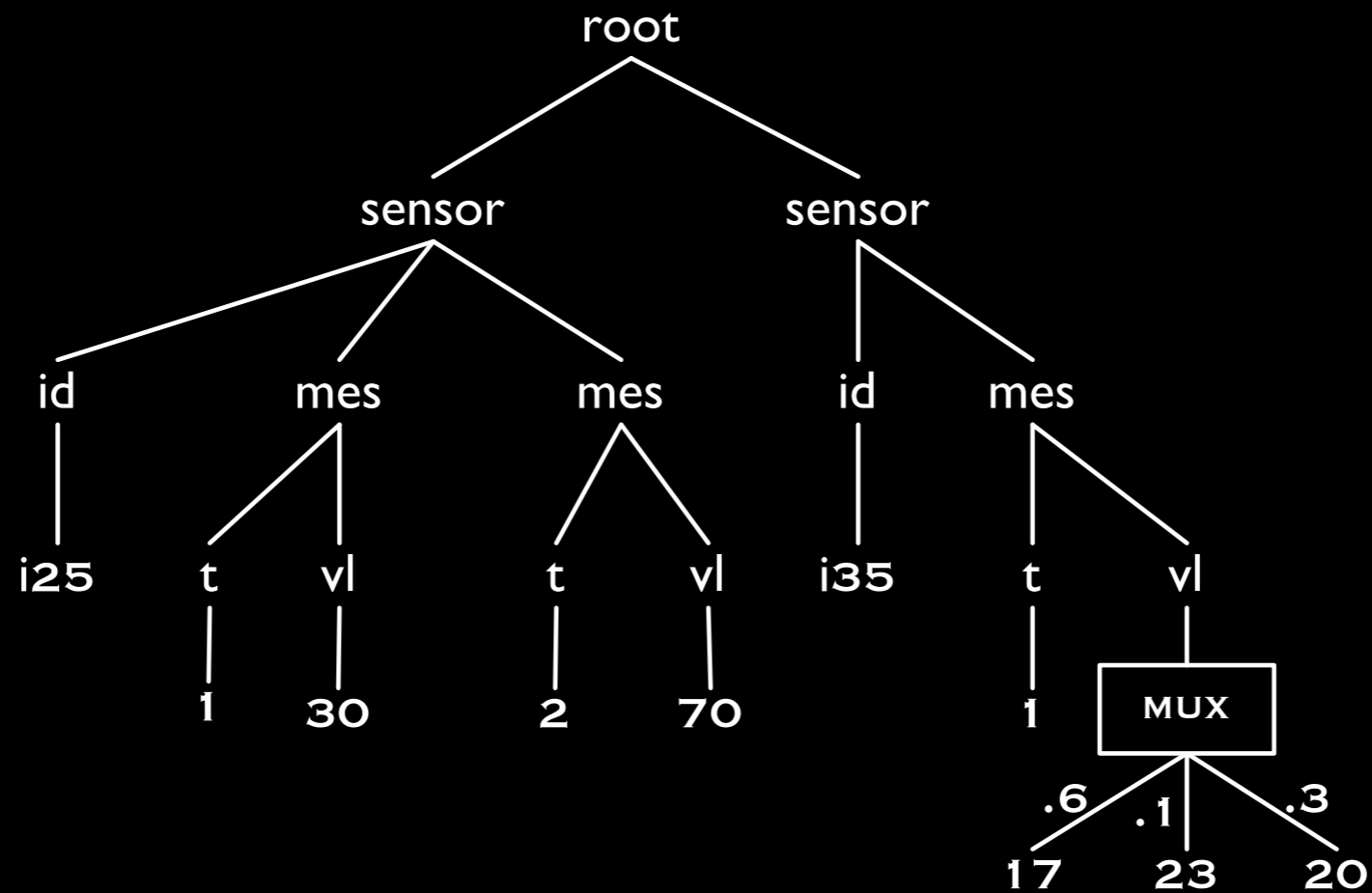


World W:

- e = true (no rain)
- MUX: 23

$$\Pr(W) = .4 \times .1$$

# Possible Worlds Semantics

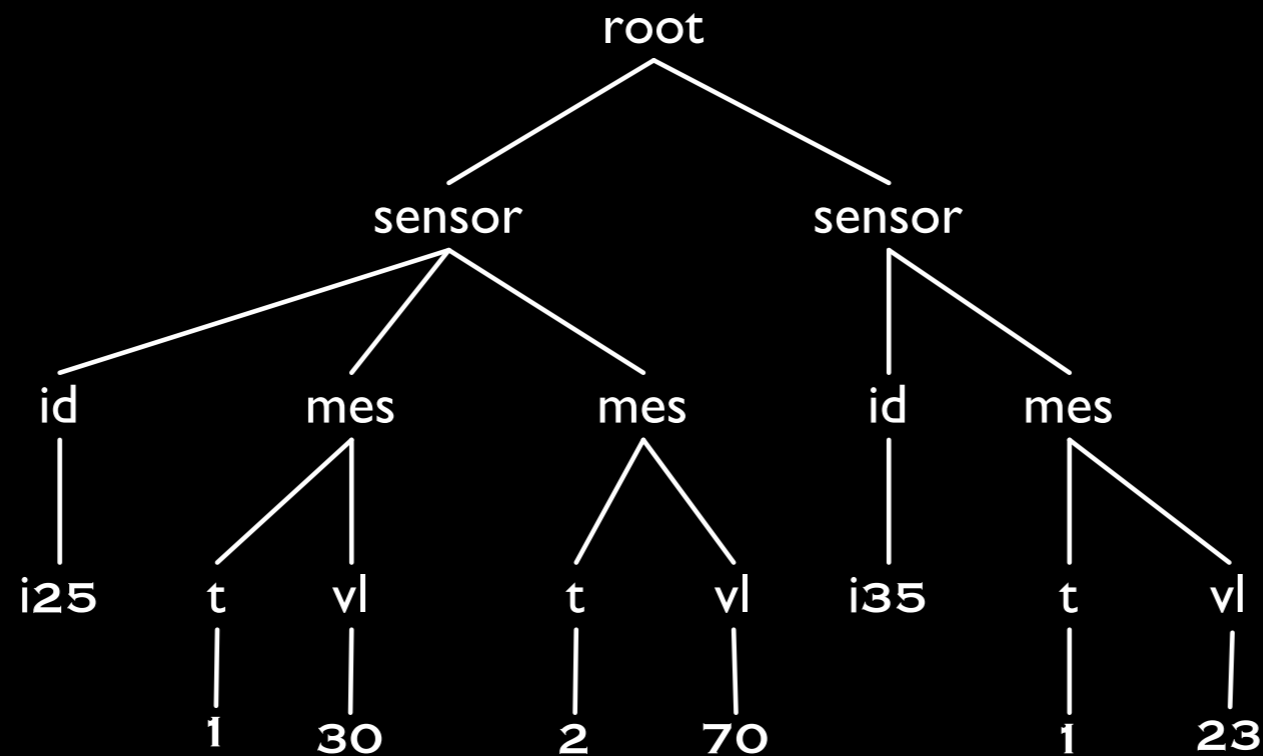


World W:

- e = true (no rain)
- MUX: 23

$$\Pr(W) = .4 \times .1$$

# Possible Worlds Semantics



World W:

- $e = \text{true}$  (no rain)
- MUX: 23

$$\Pr(W) = .4 \times .1$$

# Queries

- Was sensor i25 up at time  $t = 2$ ?
- Which sensors were up at time  $t = 2$ ?

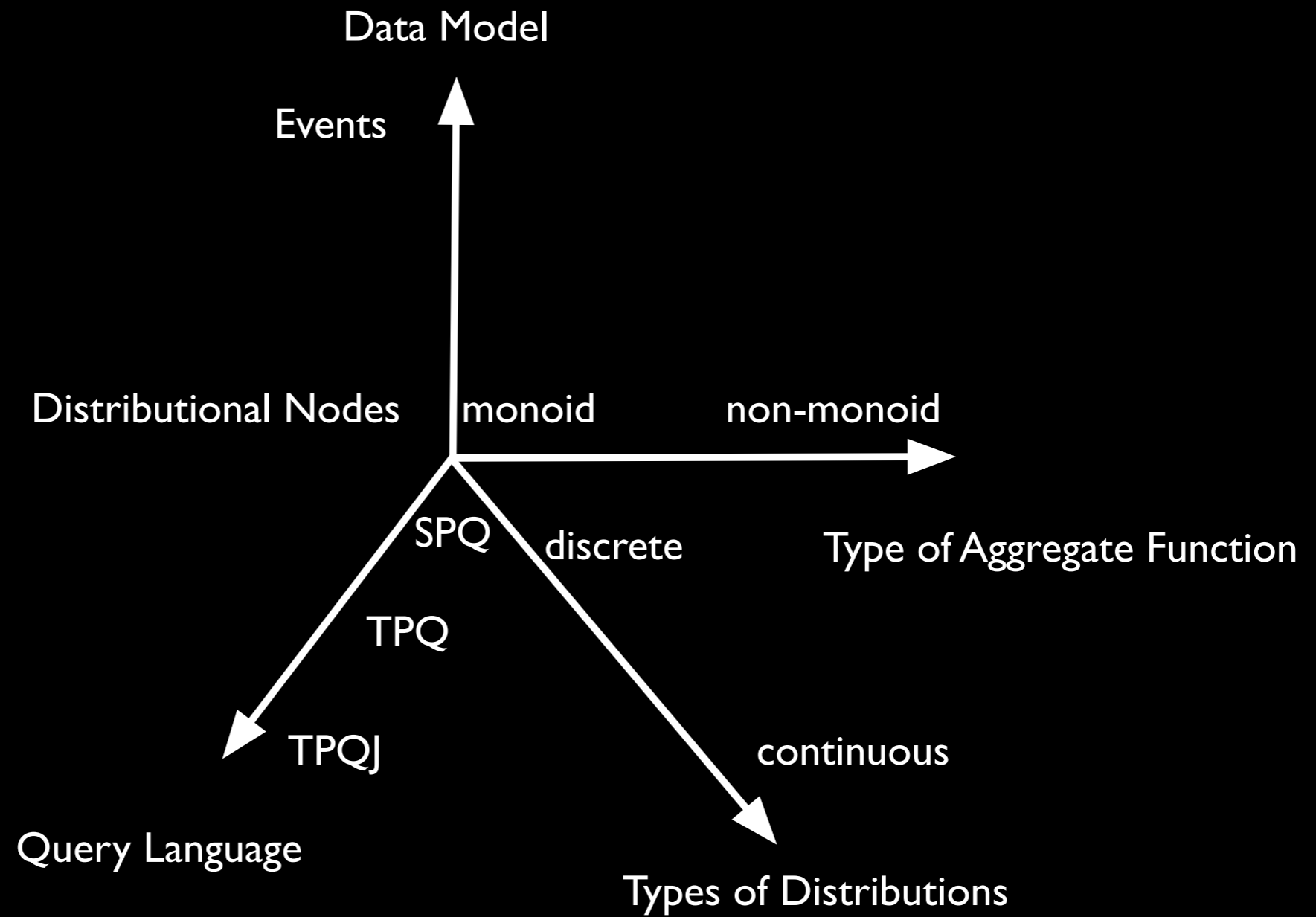
# Aggregate Queries

- How many sensors were up at time  $t = 1$ ?
- For every point in time, how many sensors were up?

⇒ we want answers to **aggregate** queries:

MIN/MAX, TopK, COUNT, SUM, COUNTD, AVG

# Global Picture



- Previous Work

# Relational Model (Re, Suciú)

<u>Item</u>	Forecaster	Amount	P
Widget	Alice	-\$99k	0.99
	Bob	\$100M	0.01
Whatsit	Alice	\$1M	1

Model: block independent PDBs,  
that is, with local probabilistic relationships (MUX)

# Relational Model (Re, Suciu)

- Queries: **conjunctive**
  - **Hierarchical** without self joins: PTIME
  - Others:  $FP^{\#P}$ -complete
- Queries: conjunctive with **aggregates** in **HAVING**  
Sufficient conditions for tractability depending on the aggregate functions
- Monte-Carlo simulations: intractable cases, Top-K

# Semi-structured Model (Kimelfeld, Senellart, ...)

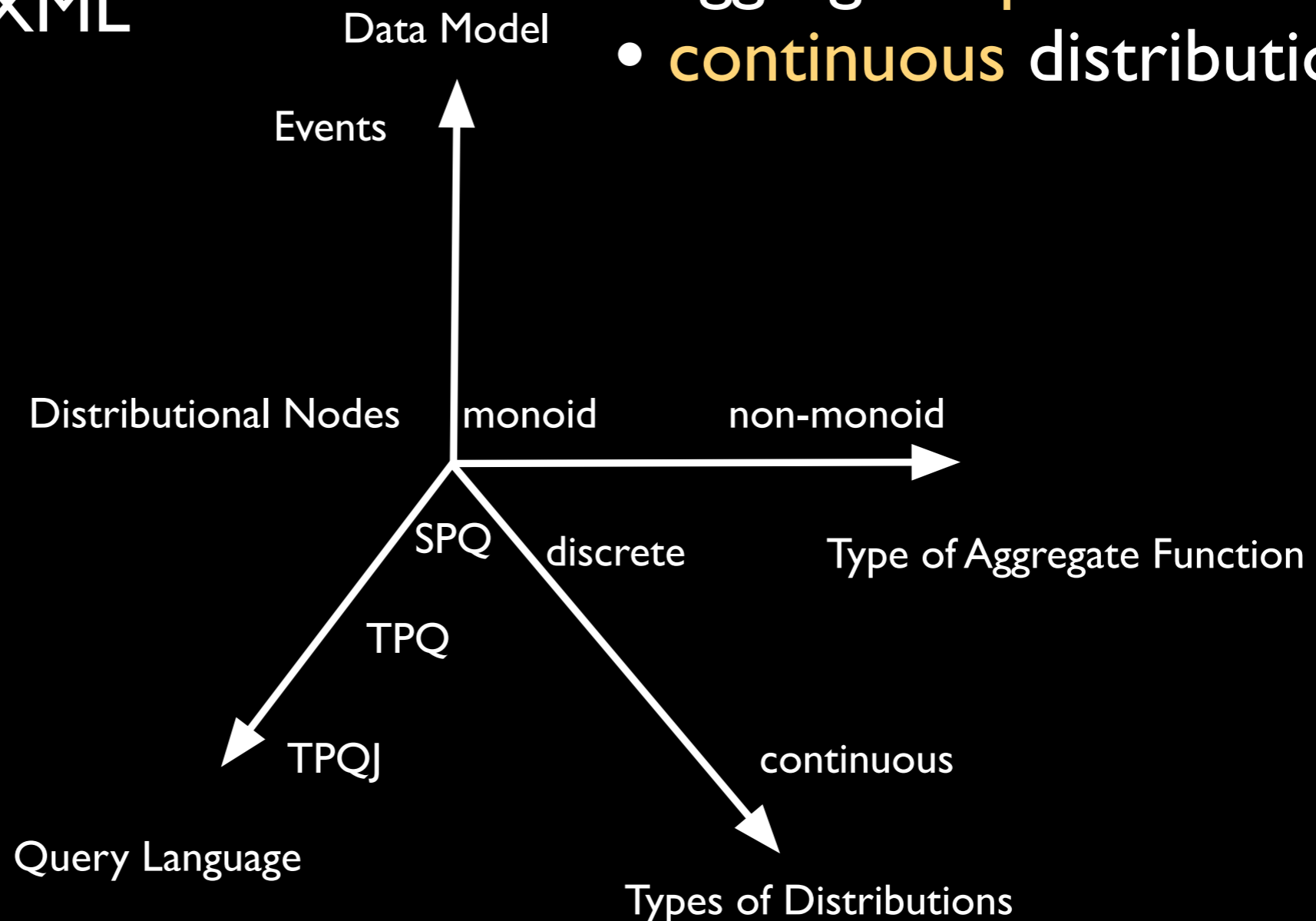
- Model: Distributional nodes + Events
- Queries: **TPQ** ~ conjunctive without real **joins**
  - distributional nodes: PTIME
  - events:  $FP^{\#P}$ -complete
- **Constraints with aggregates:**  
behave like HAVING in relational cases
- Monte-Carlo simulations for intractable cases

# Road Map

- done in relational PDB
- done in PXML
- our goal

None have considered:

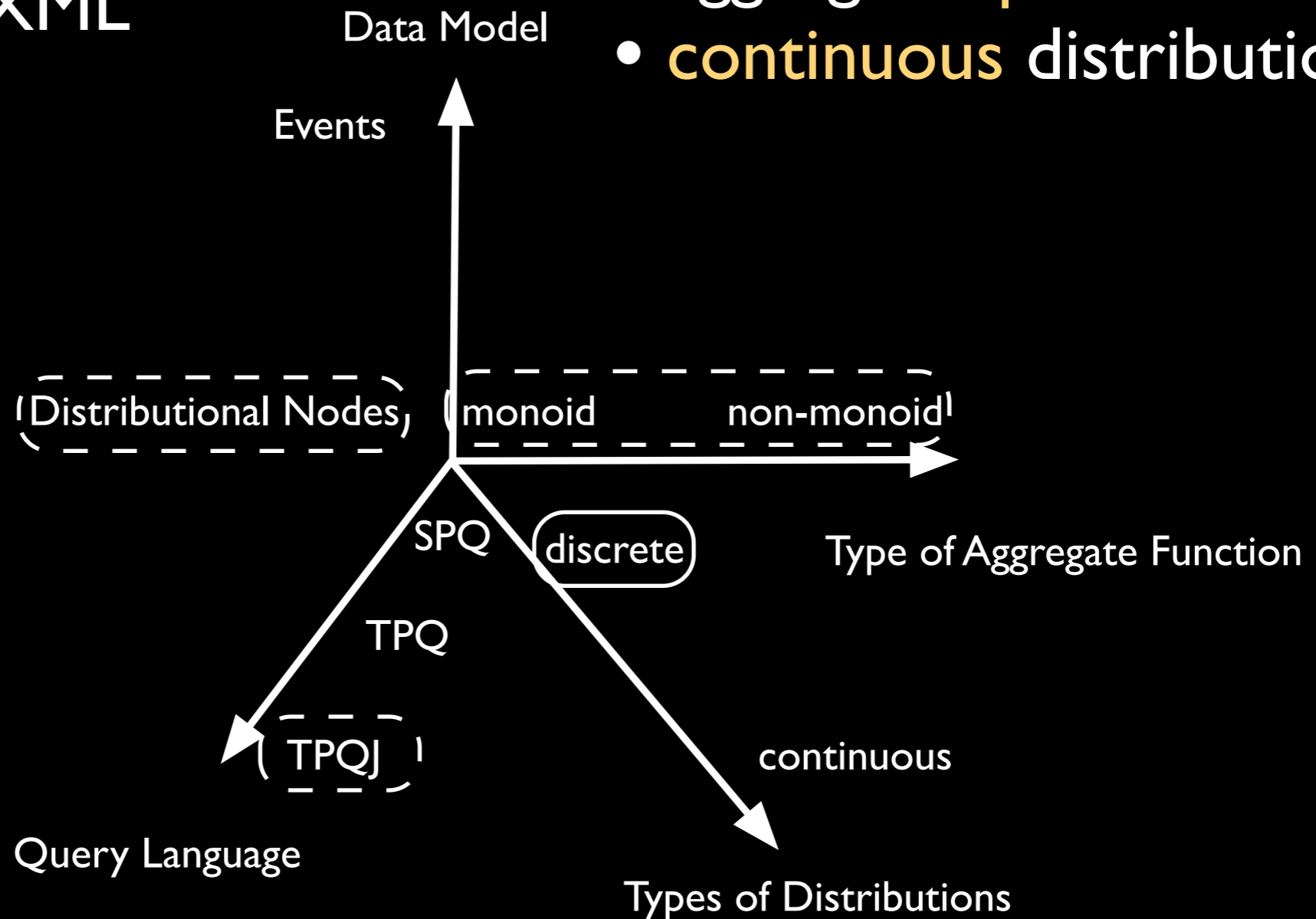
- aggregate **queries**
- **continuous** distributions



# Road Map

- done in relational PDB
- done in PXML
- our goal

- None have considered:
- aggregate queries
  - continuous distributions

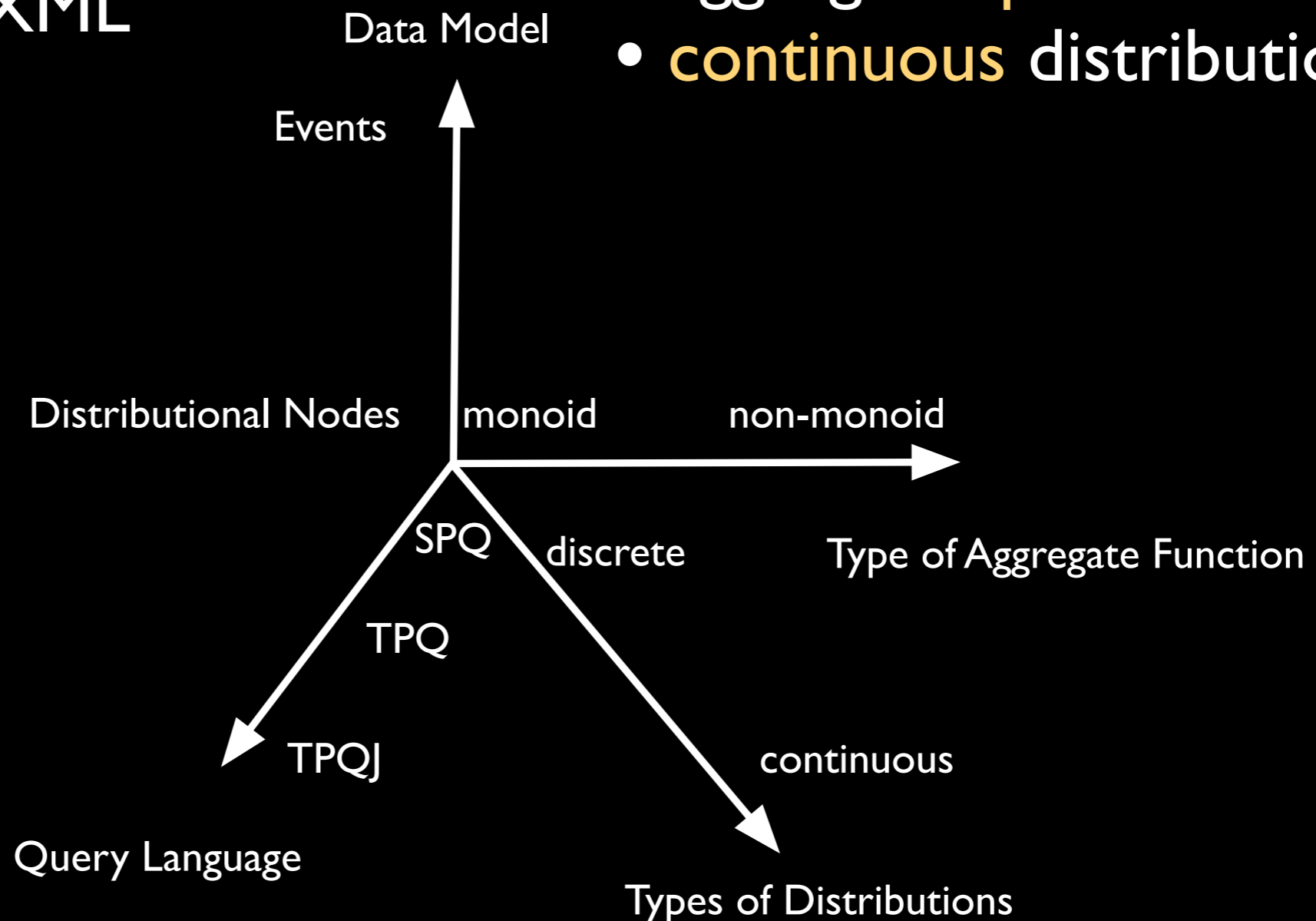


# Road Map

- done in relational PDB
- done in PXML
- our goal

None have considered:

- aggregate **queries**
- **continuous** distributions

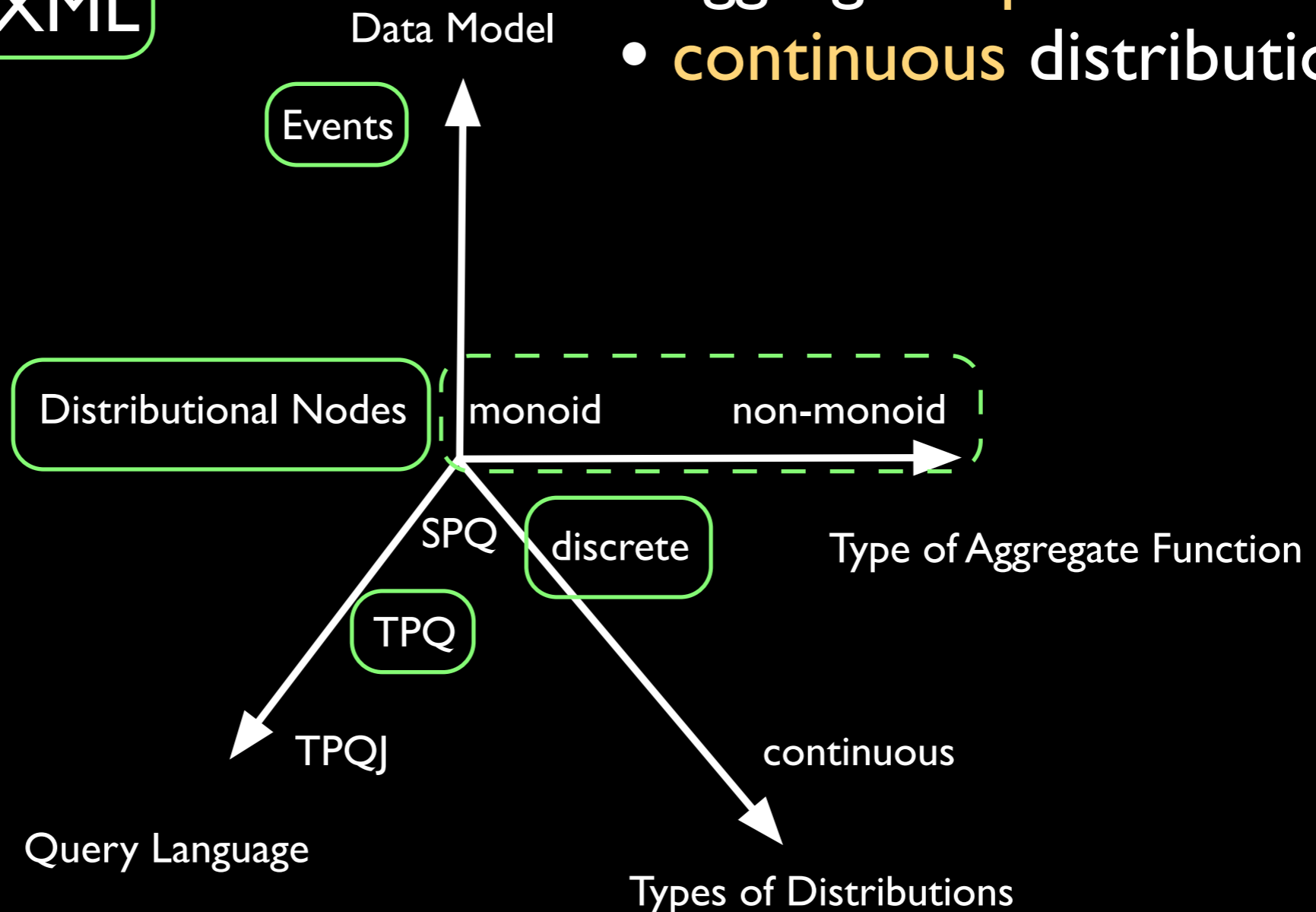


# Road Map

- done in relational PDB
- done in **PXML**
- our goal

None have considered:

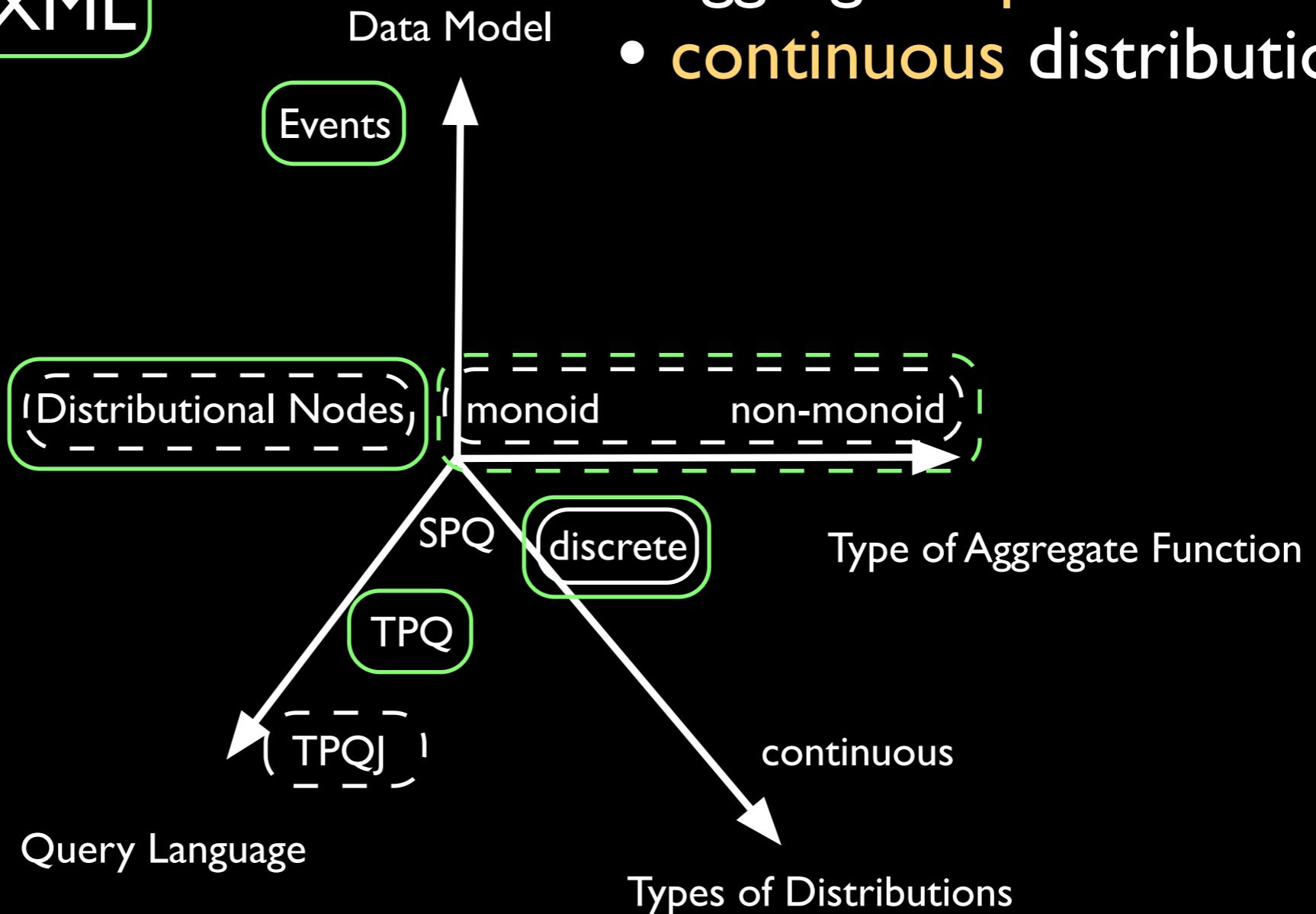
- aggregate **queries**
- **continuous** distributions



# Road Map

- done in relational PDB
- done in PXML
- our goal

- None have considered:
- aggregate queries
  - continuous distributions

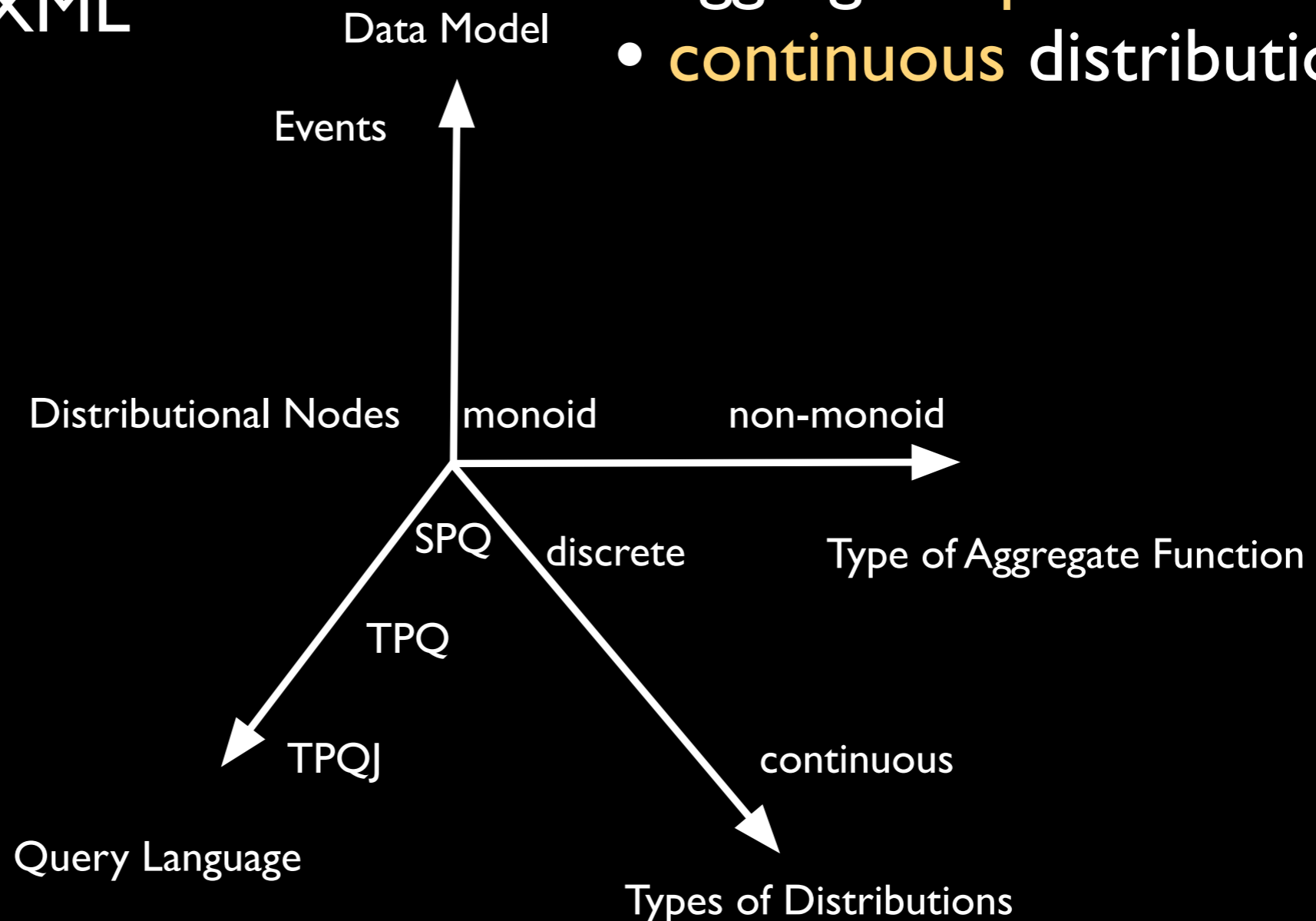


# Road Map

- done in relational PDB
- done in PXML
- our goal

None have considered:

- aggregate **queries**
- **continuous** distributions

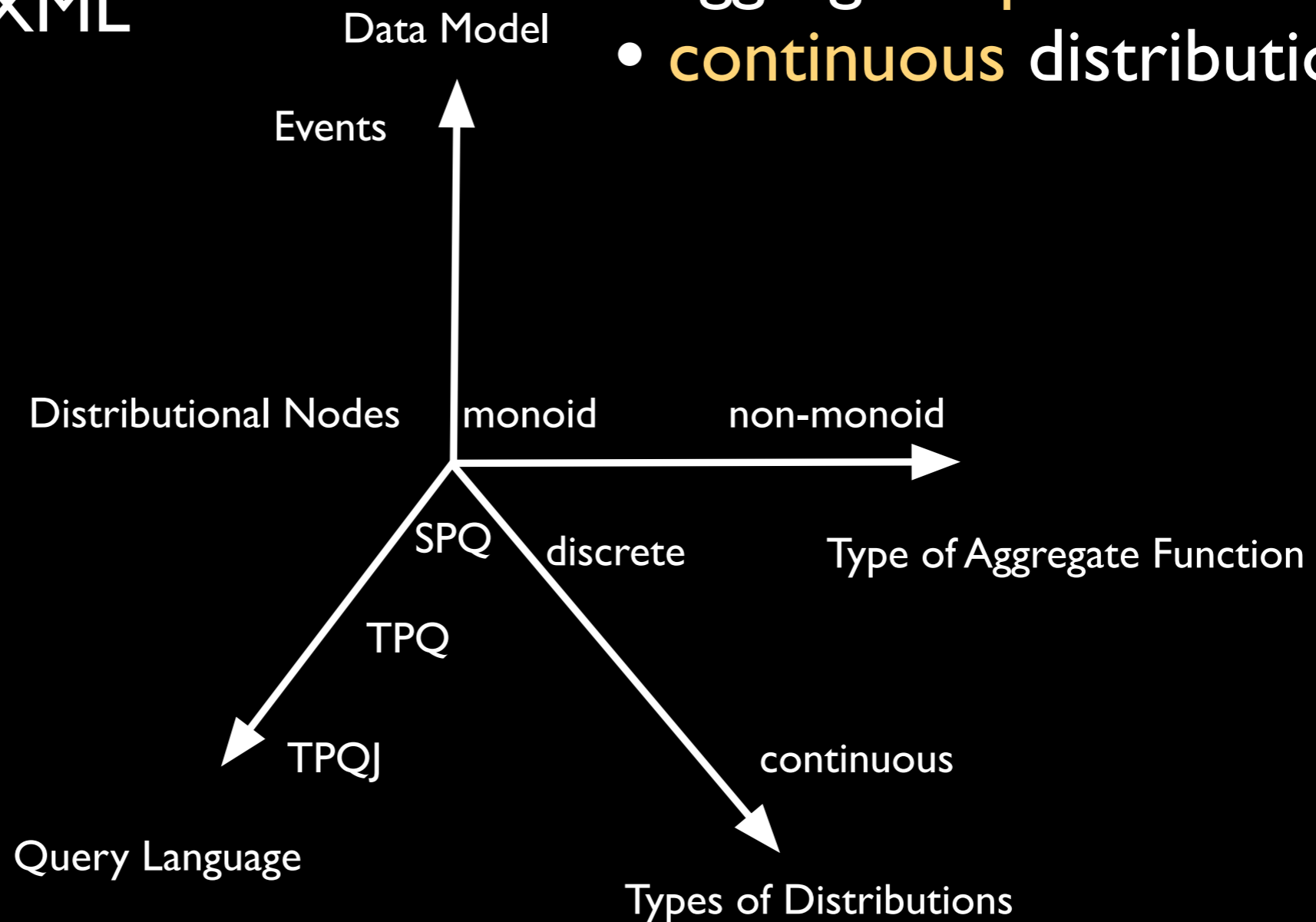


# Road Map

- done in relational PDB
- done in PXML
- our **goal**

None have considered:

- aggregate **queries**
- **continuous** distributions

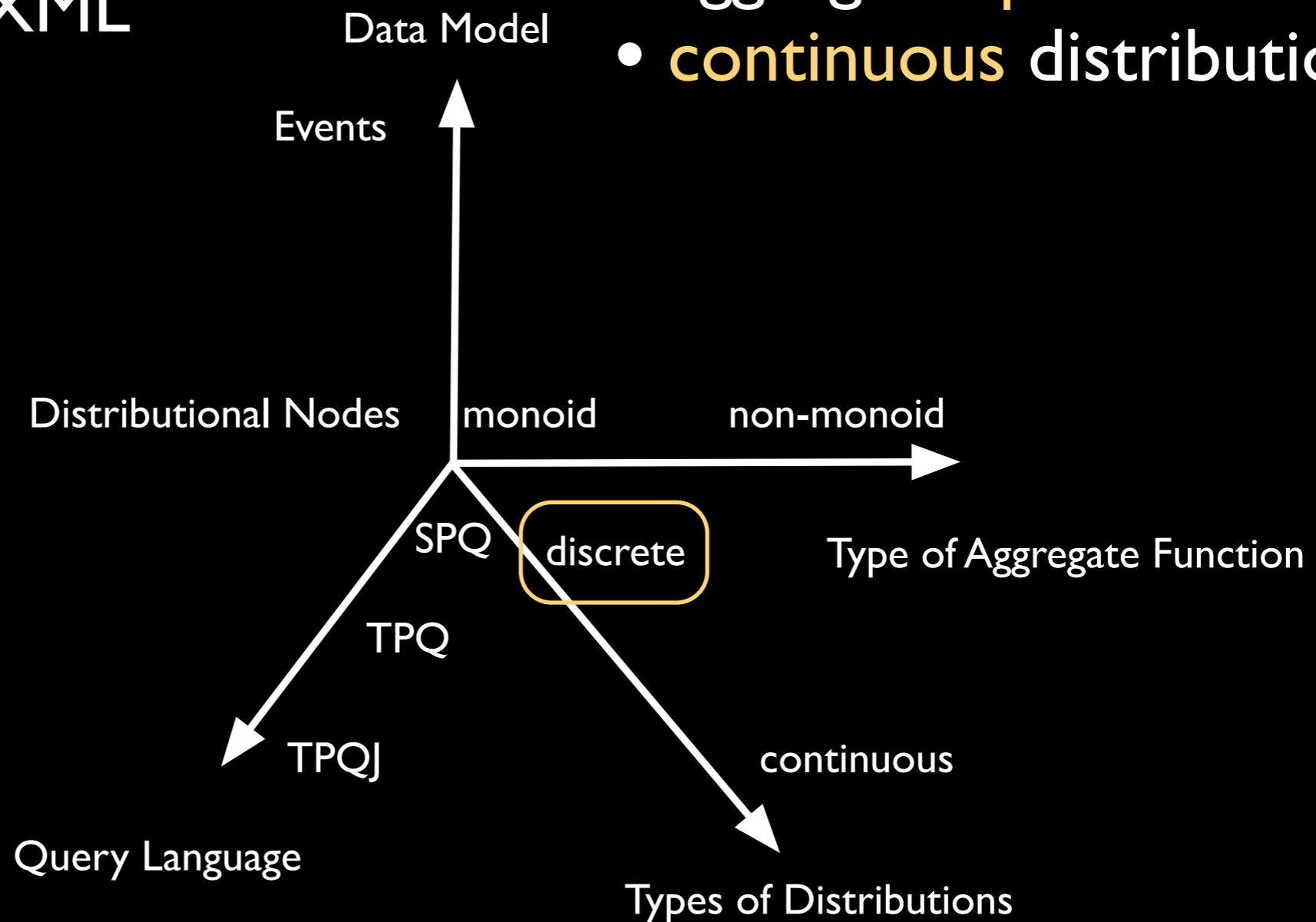


# Road Map

- done in relational PDB
- done in PXML
- our **goal**

None have considered:

- aggregate **queries**
- **continuous** distributions

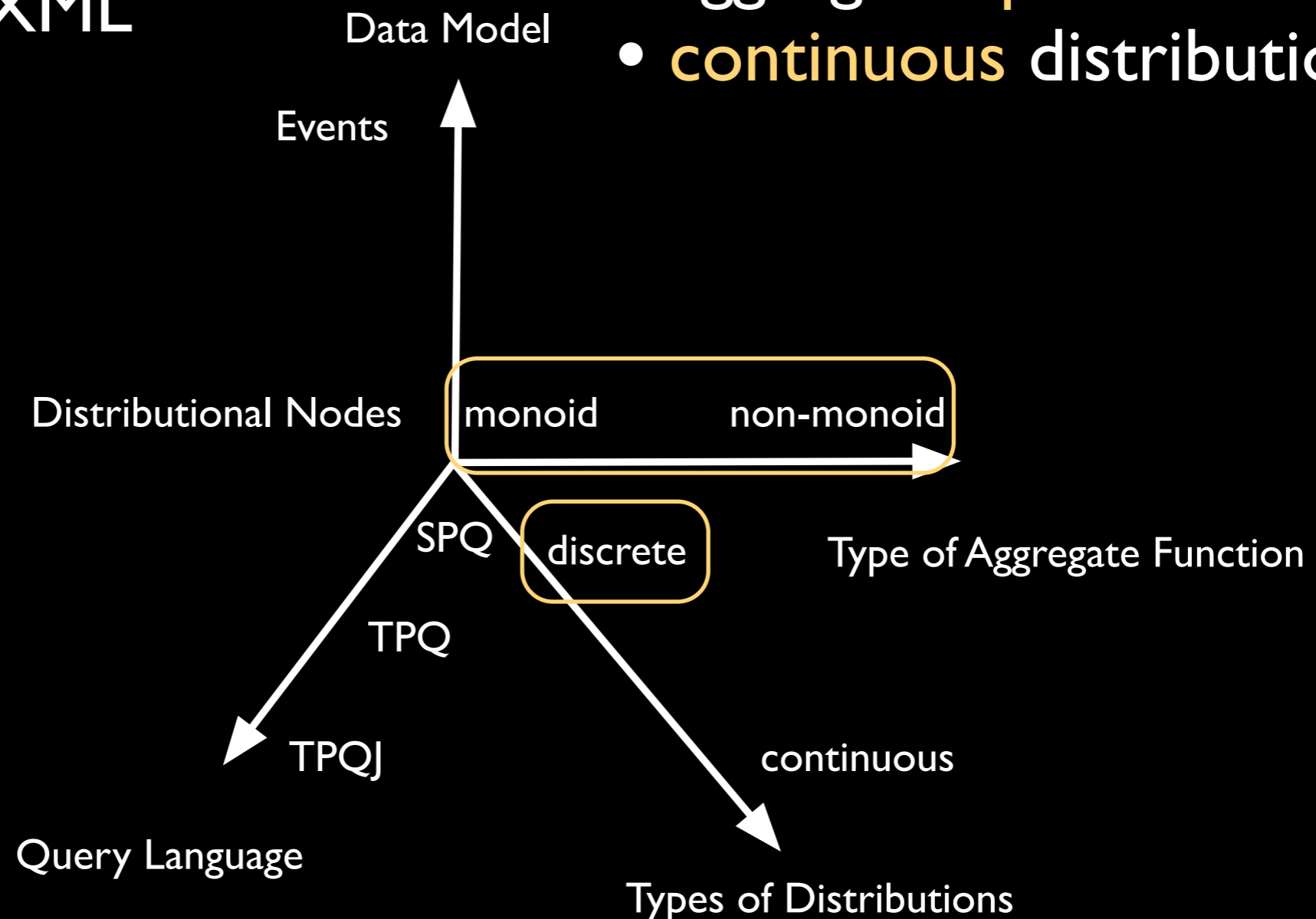


# Road Map

- done in relational PDB
- done in PXML
- our **goal**

None have considered:

- aggregate **queries**
- **continuous** distributions

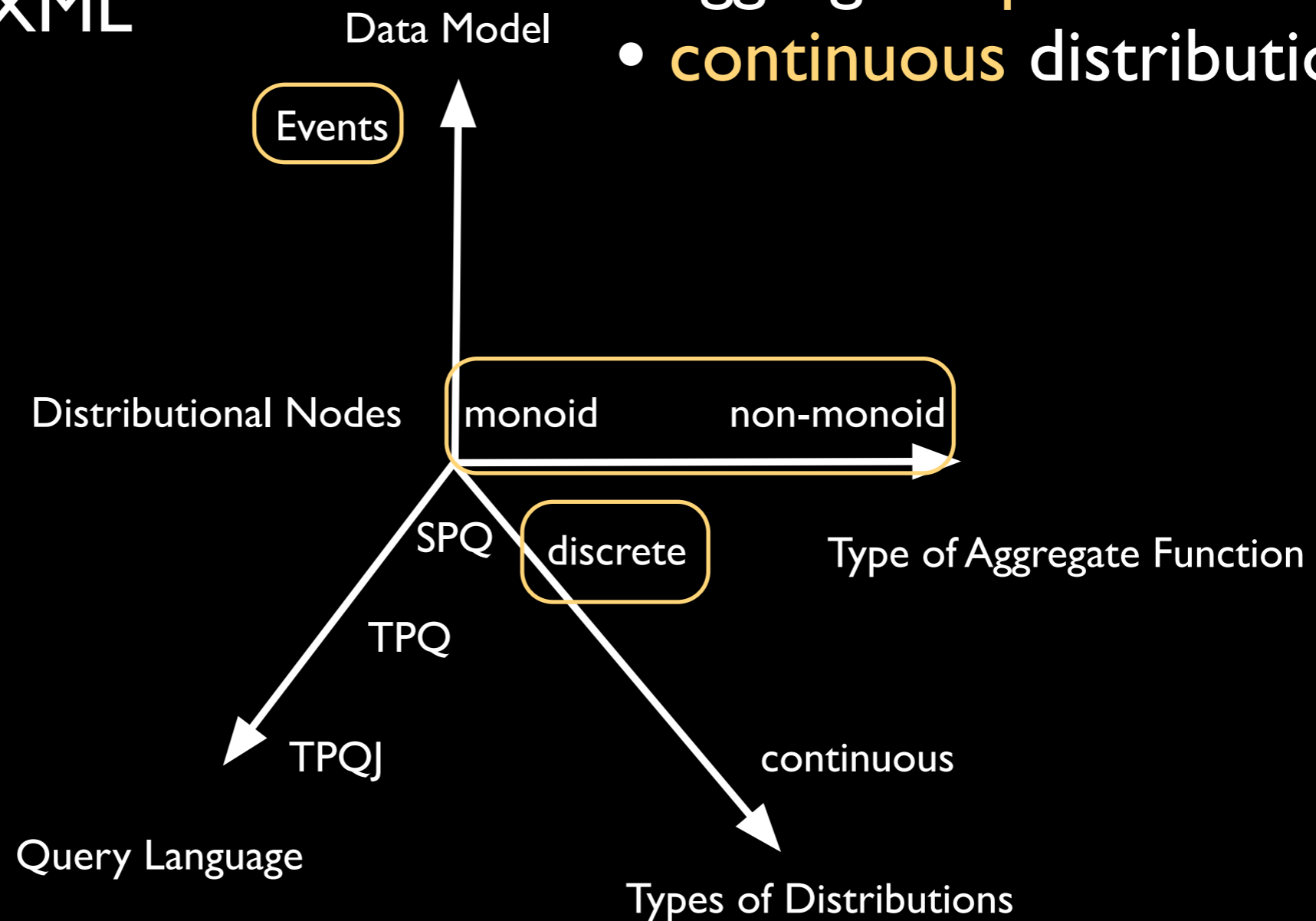


# Road Map

- done in relational PDB
- done in PXML
- our **goal**

None have considered:

- aggregate **queries**
- **continuous** distributions

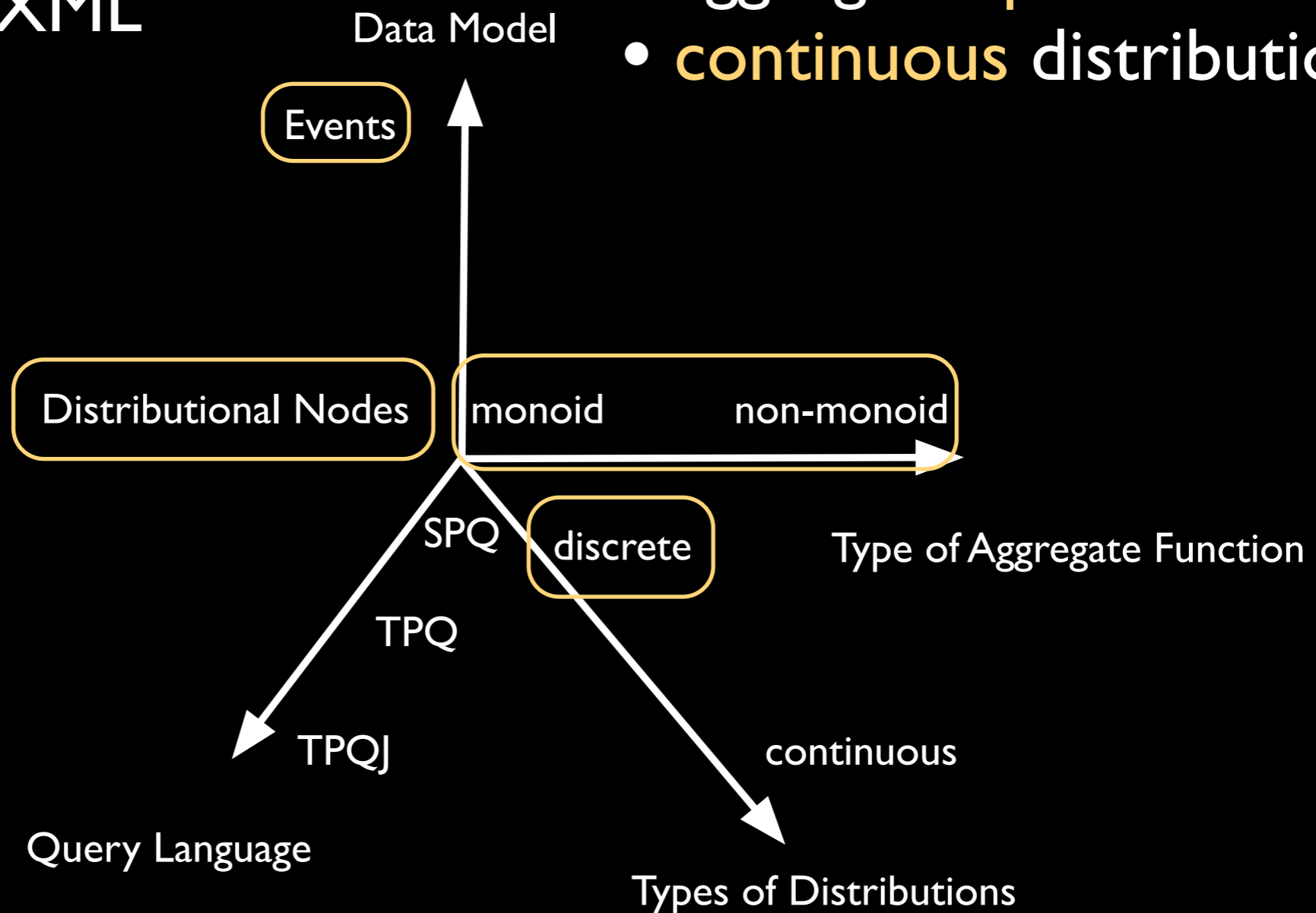


# Road Map

- done in relational PDB
- done in PXML
- our **goal**

None have considered:

- aggregate **queries**
- **continuous** distributions

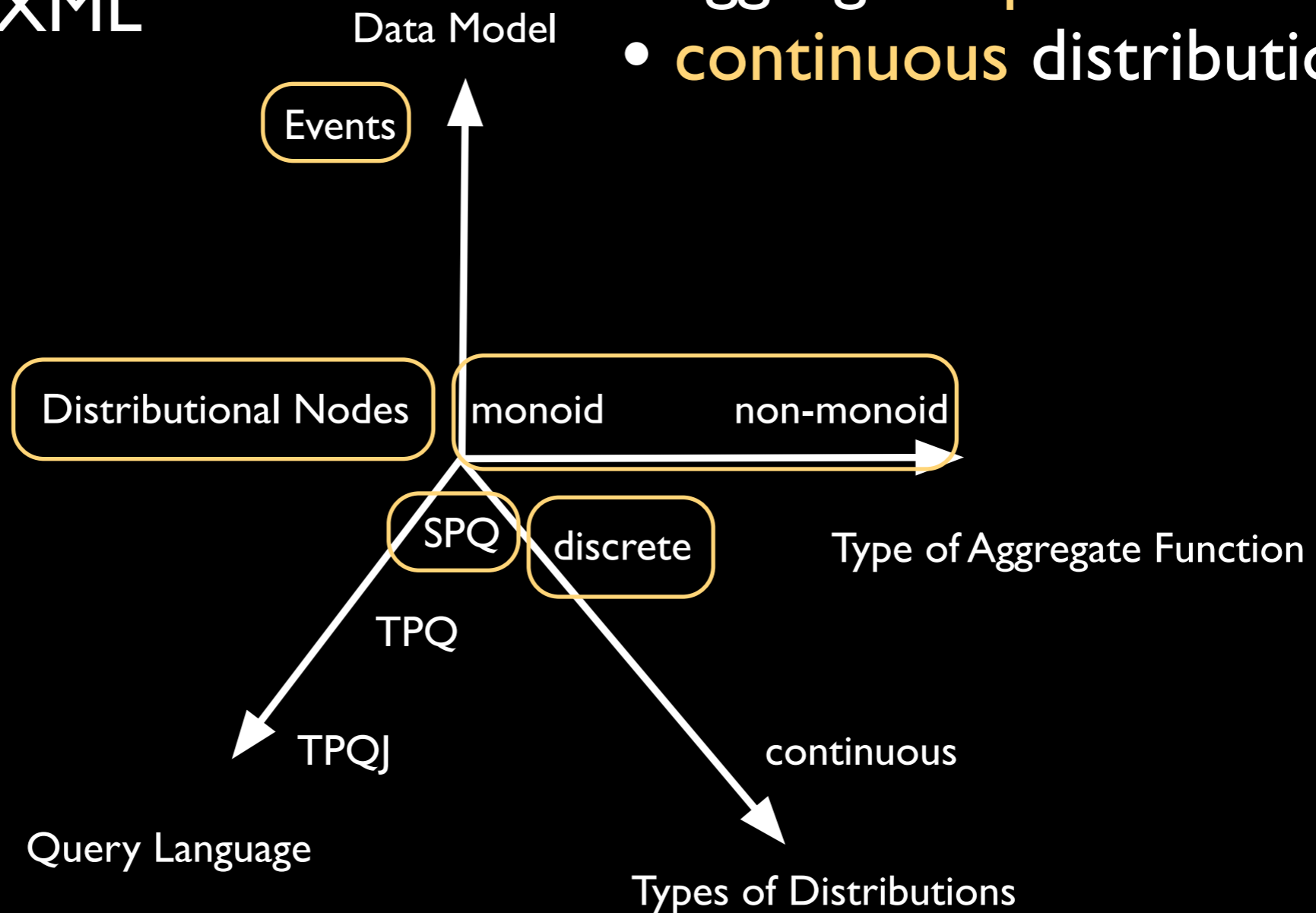


# Road Map

- done in relational PDB
- done in PXML
- our **goal**

None have considered:

- aggregate **queries**
- **continuous** distributions

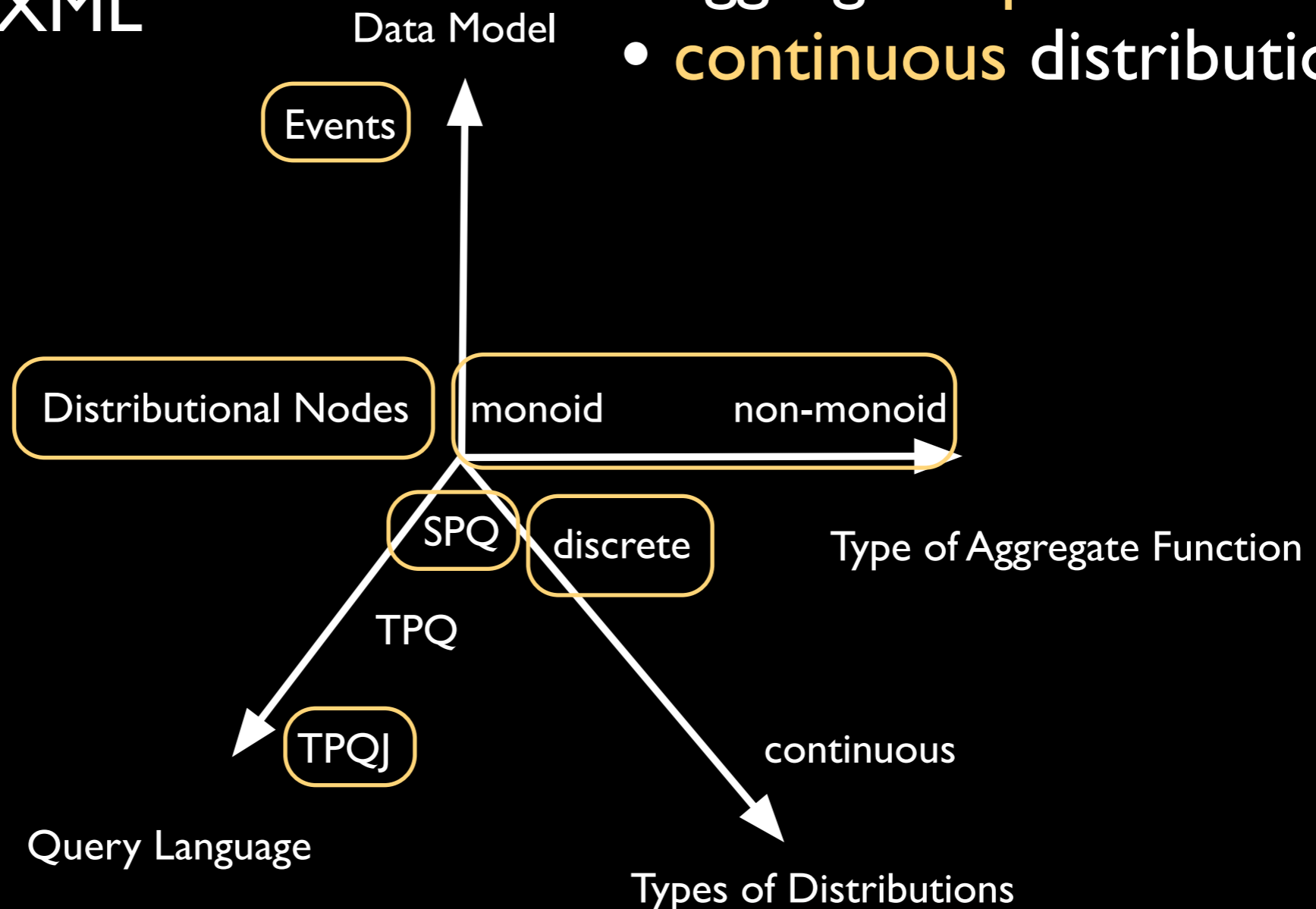


# Road Map

- done in relational PDB
- done in PXML
- our **goal**

None have considered:

- aggregate **queries**
- **continuous** distributions

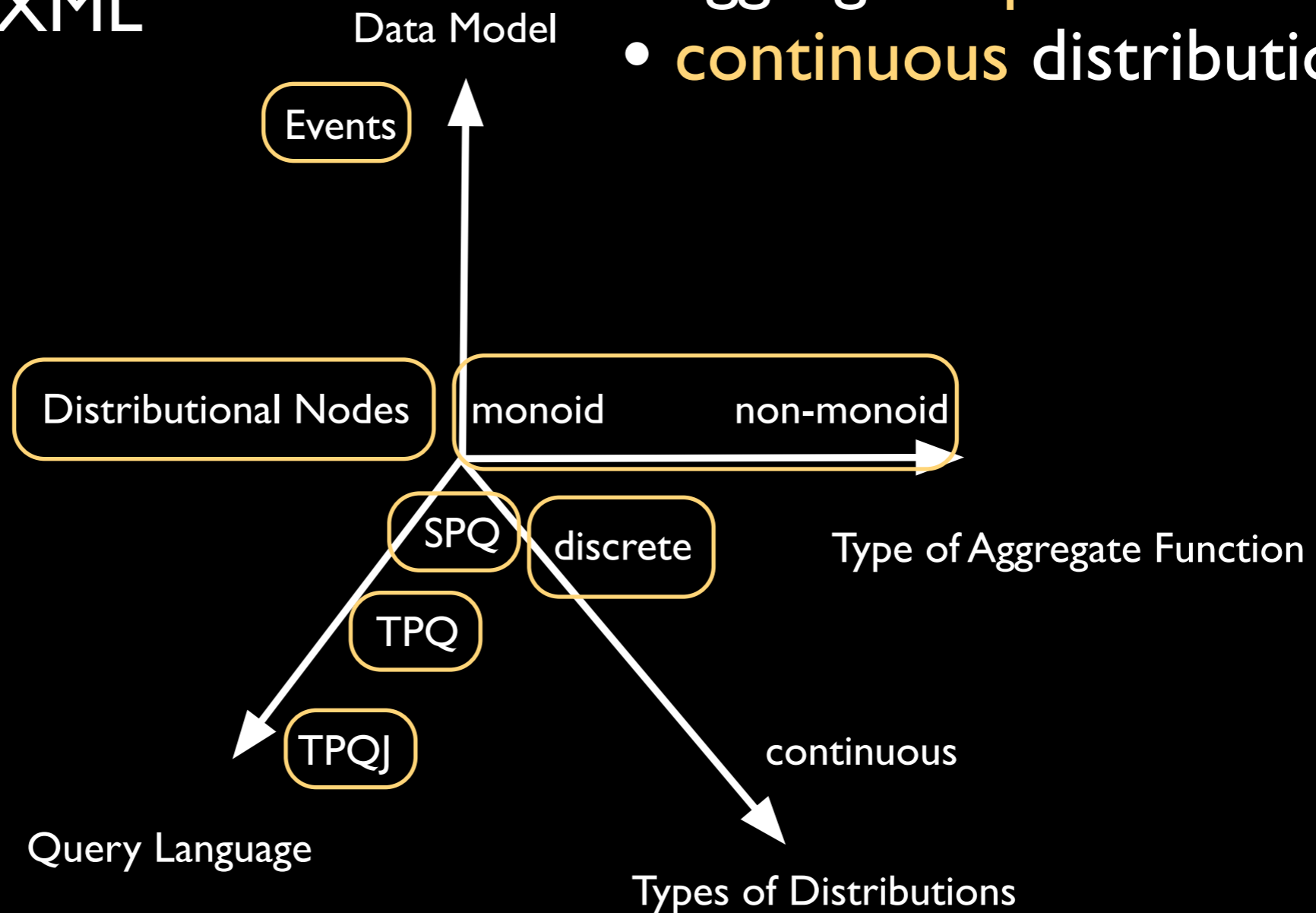


# Road Map

- done in relational PDB
- done in PXML
- our **goal**

None have considered:

- aggregate **queries**
- **continuous** distributions

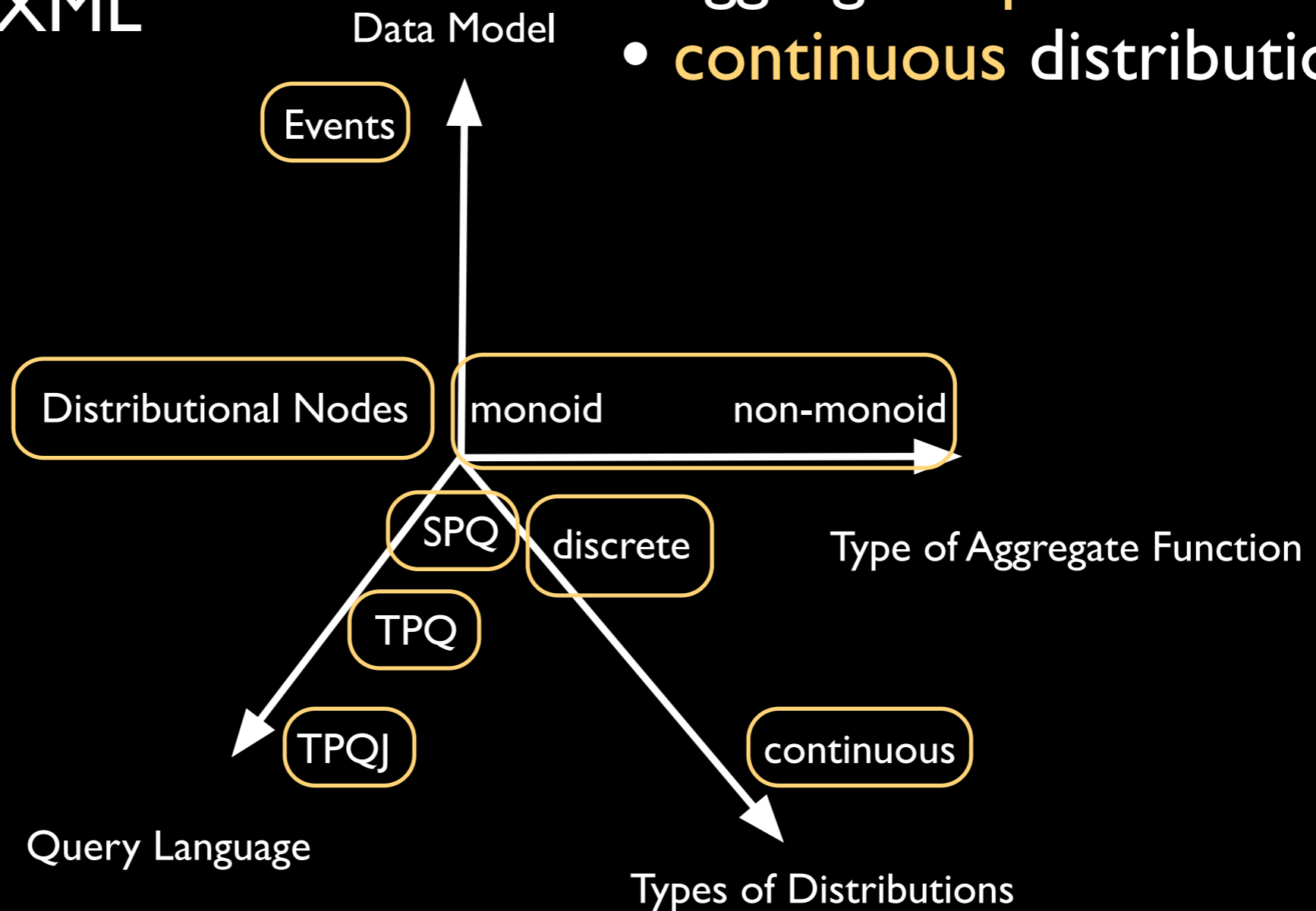


# Road Map

- done in relational PDB
- done in PXML
- our **goal**

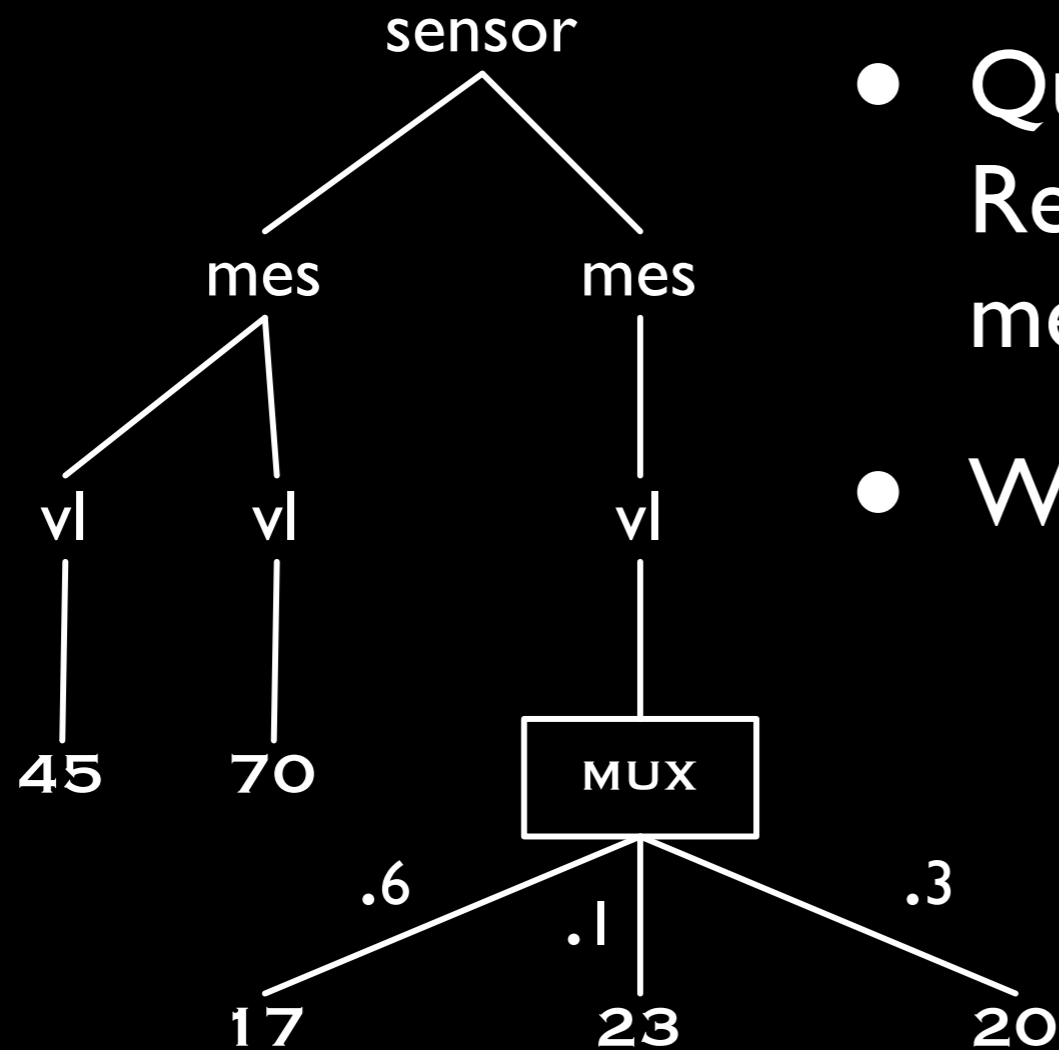
None have considered:

- aggregate **queries**
- **continuous** distributions



- Aggregating PXML without Querying

# Aggregate Functions over PXML



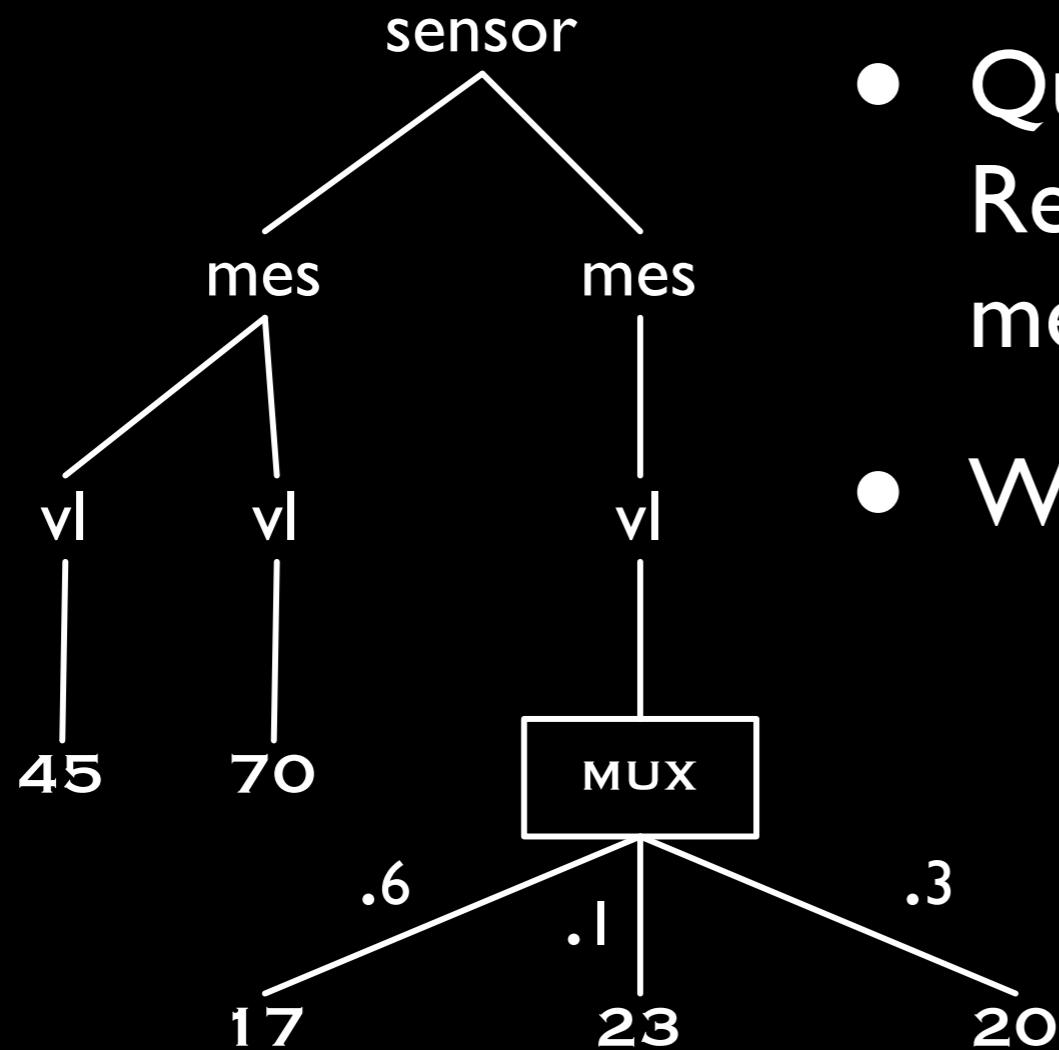
- Query:  
Return the AVG of the measurements
- What should be an answer?

$$\text{AVG}(W17) = 44, P=.6$$

$$\text{AVG}(W23) = 46, P=.1$$

$$\text{AVG}(W20) = 45, P=.3$$

# Aggregate Functions over PXML



- Query:  
Return the AVG of the measurements
- What should be an answer?

$$\text{AVG}(W17) = 44, P=.6$$

$$\text{AVG}(W23) = 46, P=.1$$

$$\text{AVG}(W20) = 45, P=.3$$

**Distribution** of the aggregate values over all worlds

# Aggregate Functions over PXML

- Aggregate functions AGG = random variables
- $AGG(D) = \text{distribution}$  of the aggregate values over  $Worlds(D)$
- Functions of our interest:  
COUNT, SUM, MIN, MAX, COUNTD, AVG

# Problems to Investigate

For PXML document  $D$ , constant  $C$

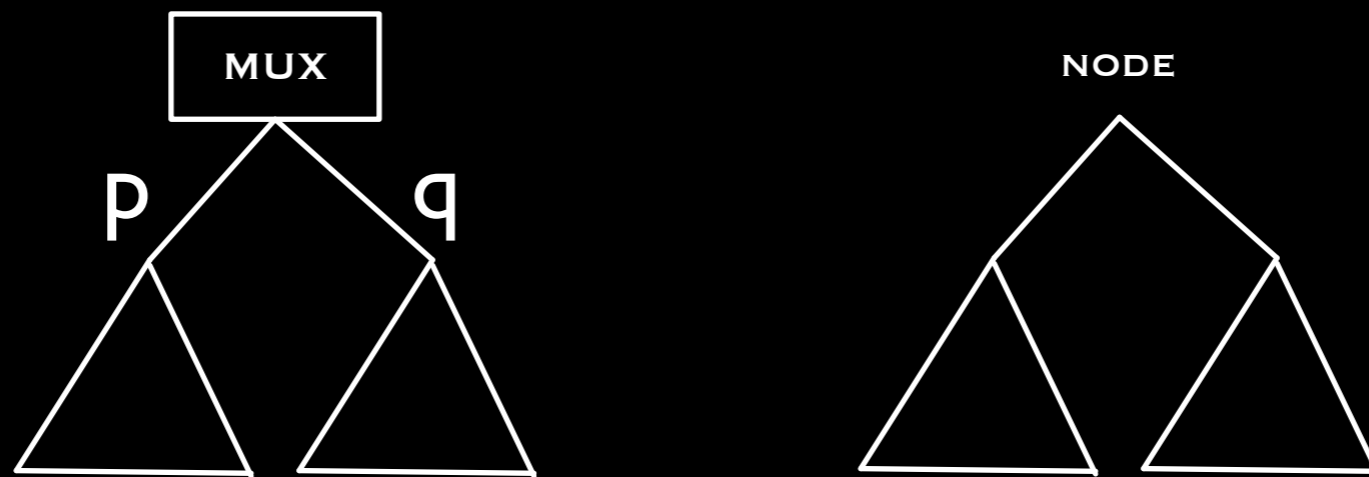
- **Membership:**  
decide  $\Pr(\text{Agg}(D)=C) > 0$
- **Probability computation:**  
compute  $\Pr(\text{Agg}(D)=C)$
- **Moment computation:**  
compute  $E(\text{Agg}(D)^k)$

# Aggregating PXML-Events

- Almost all problems are **intractable** for MIN, MAX, COUNT, SUM, COUNTD, AVG:
  - Membership is **NP-complete**
  - Probability and moments computation **FP<sup>#P</sup>-complete**
- **Polynomial** cases:  
**moments** computation for SUM and COUNT

- Aggregating PXML-MUX

# Hierarchical Structural Property of PXML-MUX

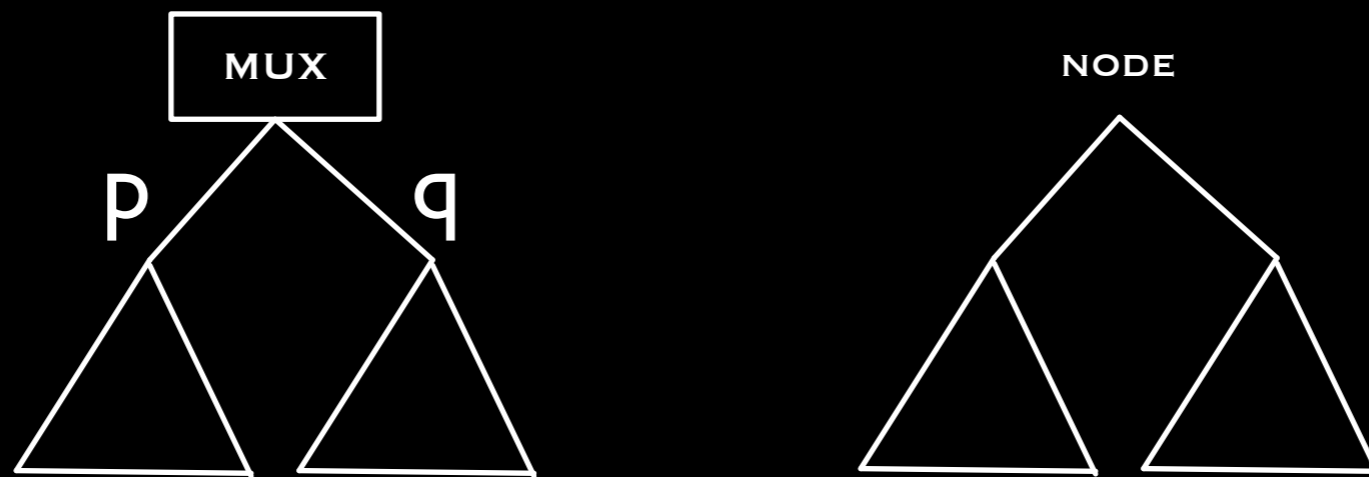


What can we say about PXML-MUX documents?

Composed from:

- **independent**
- **hierarchically** organized probabilistic components

# Hierarchical Structural Property of PXML-MUX



What can we say about PXML-MUX documents?

- All probabilistic dependences are local
- **distribution** for composed p-document  
= **composition** of distributions of the components  
with **convex sums** and **convolutions**

# Some Aggregate Functions are Easier than Others

Functions that admit **divide and conquer** strategy

- $SUM \{ | a, b, c, d | \} = SUM \{ | a, b | \} + SUM \{ | c, d | \}$
- They are called **monoid** aggregation function:  
Let  $M = (A, *)$  be a monoid  
 $Agg \{ | a, \dots, n | \} = Agg \{ | a | \} * \dots * Agg \{ | n | \}$
- Monoid functions are: COUNT, SUM, MIN, MAX

# Aggregating PXML-MUX with Monoid Functions

- Monoid aggregate functions can exploit the hierarchical structure of PXML-MUX
- Most problems for monoid aggregates are **PTIME**
- Computing the distribution can be polynomial in the distribution (output) size (e.g. sum)

# Aggregating PXML-MUX with Non Monoid Functions

Non-monoid COUNTD and AVG are **hard**

- membership is in **NP**
- probability computation  **$FP^{\#P}$ -complete**
- moments computation is **PTIME**

# Approximating Query Answers

- Many problems are NP- or FP#P-complete
- There are efficient **Monte-Carlo** approximation techniques for all the problems
- For example: given Epsilon and Delta with polynomially many samples one can compute the estimation  $X$  such that
$$| P(\text{AGG}(D) = C) - X | > \text{Epsilon}$$
holds with probability at most Delta

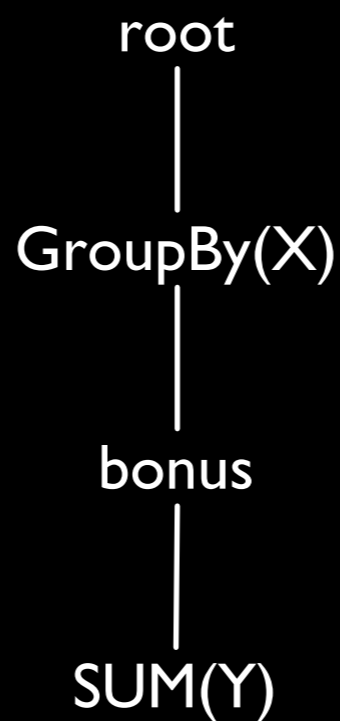
- Querying PXML

# Aggregate Queries for PXML

- Single-path TPQ with Group-By
- TPQ without joins with Group-By
- TPQ with joins and Group-By

# Single-Path TPQ with Group-By

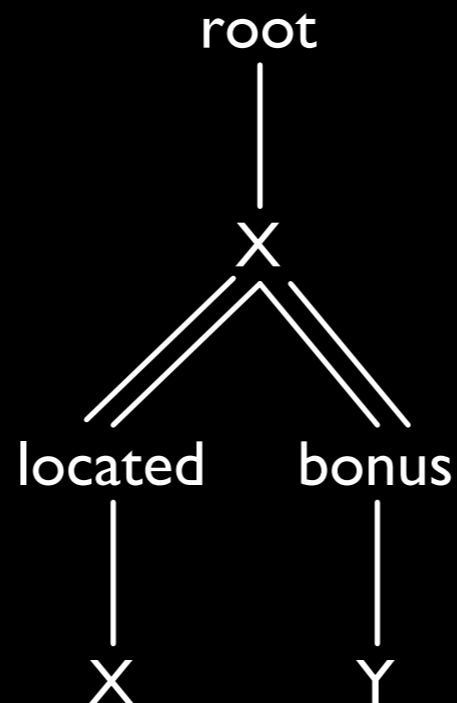
Query: Sum bonuses for every team



Query answering is **reducible** to  
aggregate function computation

# TPQ with Joins

Query: Return the bonuses,  
where the team is named by the city

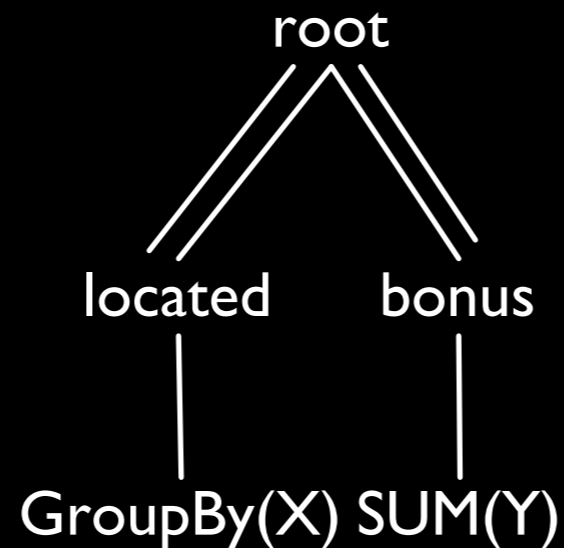


Query answering without aggregation is  
already  $FP^{\#P}$ -complete

$\Rightarrow$  **no chance** for tractable aggregate queries

# TPQ without Joins with Group-By

Query: Return the sum of bonuses per city



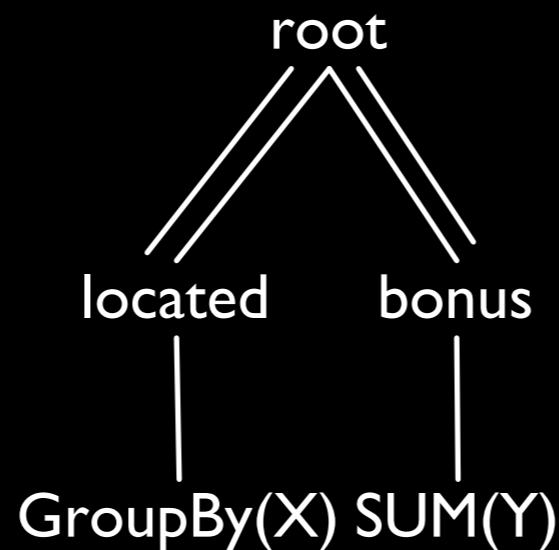
## PXML-Events:

Query answering without aggregation is  
already  $FP^{\#P}$ -complete

⇒ **no chance** for tractable aggregate queries

# TPQ without Joins with Group-By

Query: Return the sum of bonuses per city

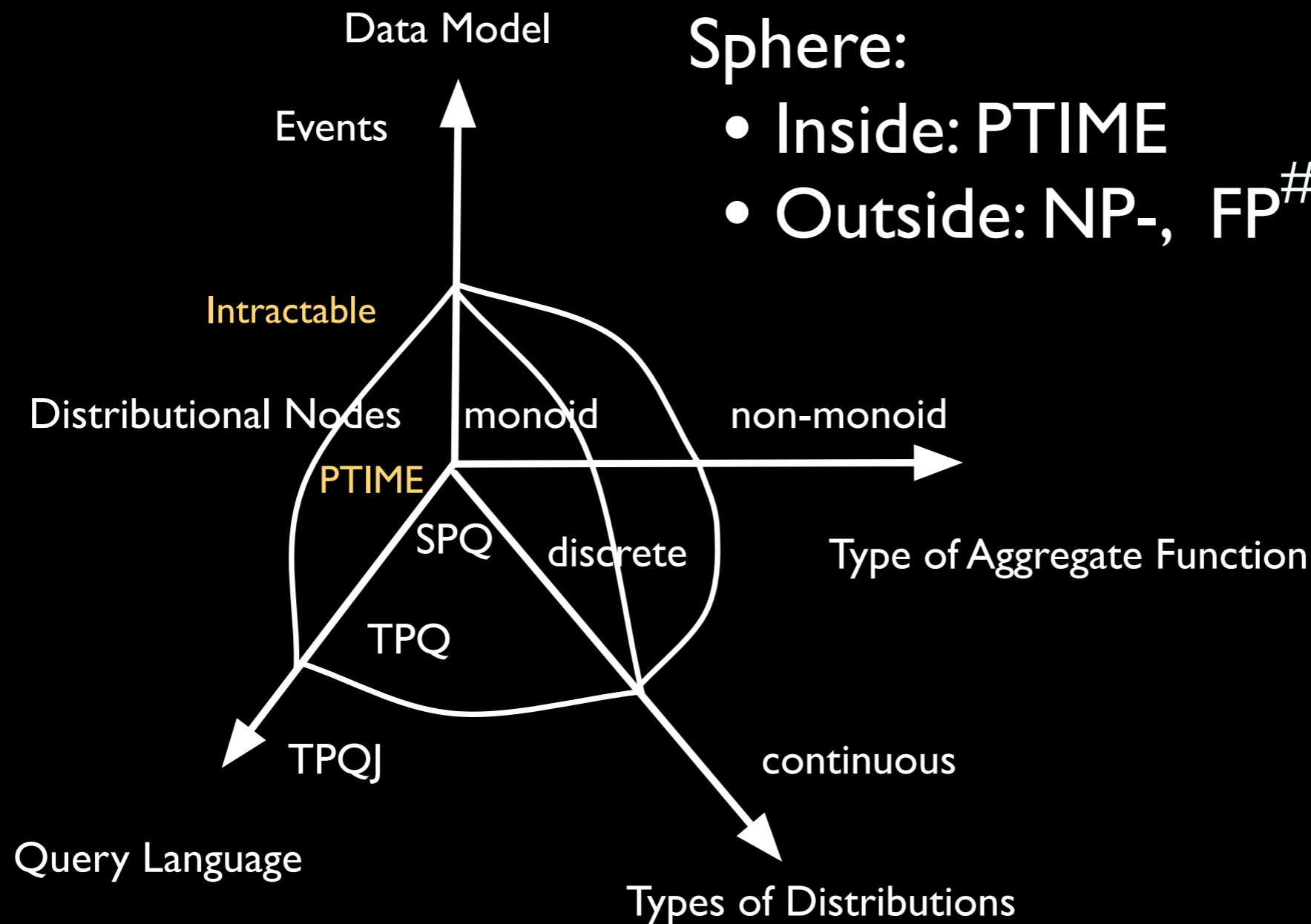


## PXML -MUX:

- COUNT, MIN, MAX, RATIO: **P**TIME [Kimelfeld at al.]
- SUM: **NP-hard** [Kimelfeld at al.]
- AVG, COUNTD : **FP<sup>#P</sup>-complete**

It holds for simple aggregation already

# Road Map: What is Done



Sphere:

- Inside: PTIME
- Outside: NP-,  $FP^{\#P}$ -complete

# Continuous Models

- **Model:** PXLM-MUX-Events and continuous distributions of the leaves' values
- What is new?
  - values distributed continuously  
⇒ SUM, etc. are **continuous** as well
  - Mixture of discrete and continuous: discrete fragments are handled with **Dirac** distributions

# Monoid Aggregates for PXML-MUX

- How to compute?
  1. Compute aggregation distributions on the leaves
  2. Push distributions bottom-up combining them with **convolutions** and **convex sums**
- It works when distribution on the leaves are closed under convolutions and convex sum
  - piecewise polynomials (for SUM, MIN/MAX)
  - Gaussian distributions (for SUM)

# Next Steps

- Dig in the continuous case
- Get better understanding of relationships between relational and semi-structured models
- Implement a system