Differential Privacy in querying RDF Knowledge Graphs

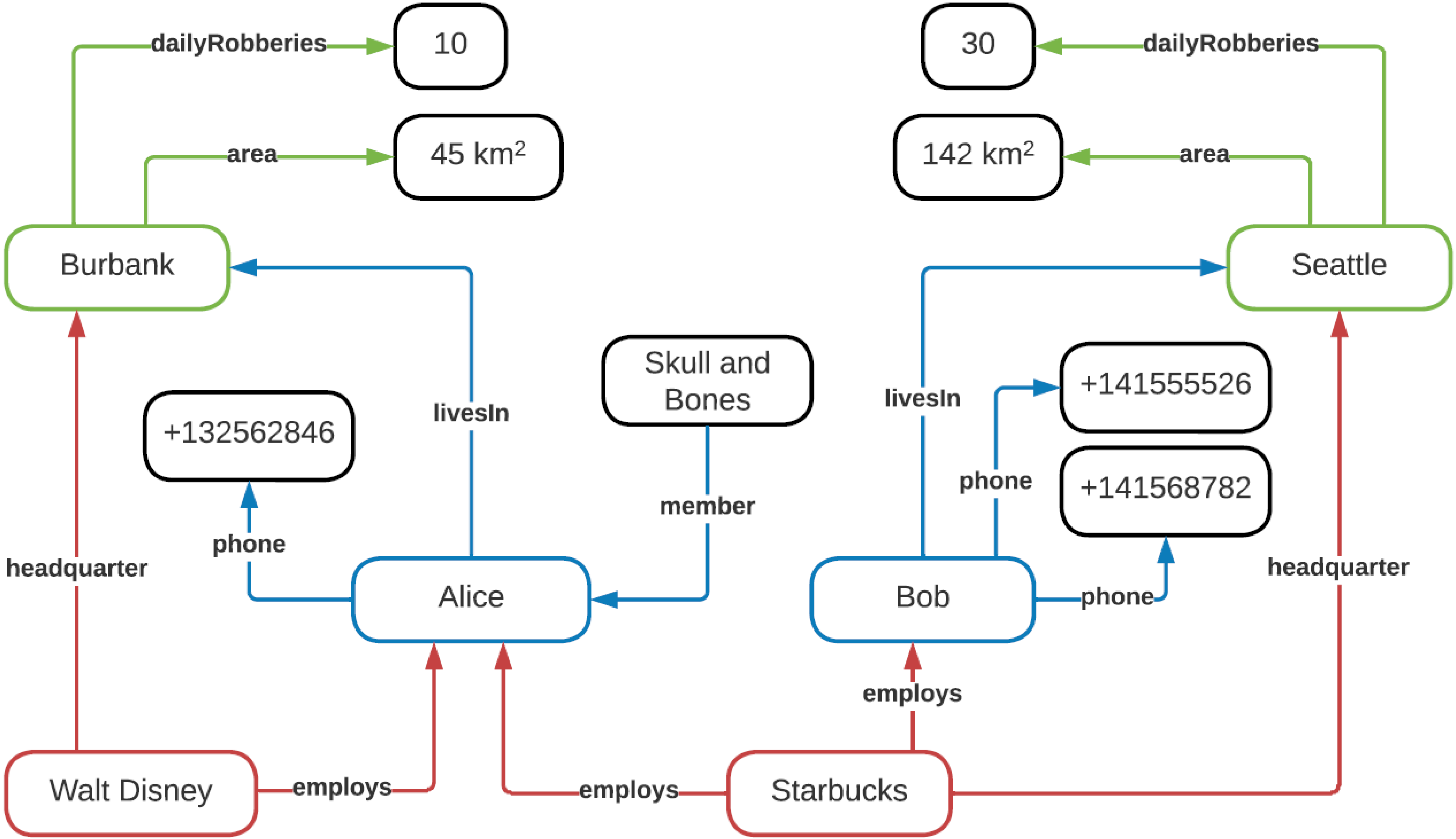
Jorge Lobo
ICREA & UPF
Barcelona, Spain

Carlos Buil Aranda
Universidad Técnica Federico Santa María and IMFD, Valparaiso, Chile

Federico Olmedo
Universidad de Chile & IMFD
Santiago, Chile

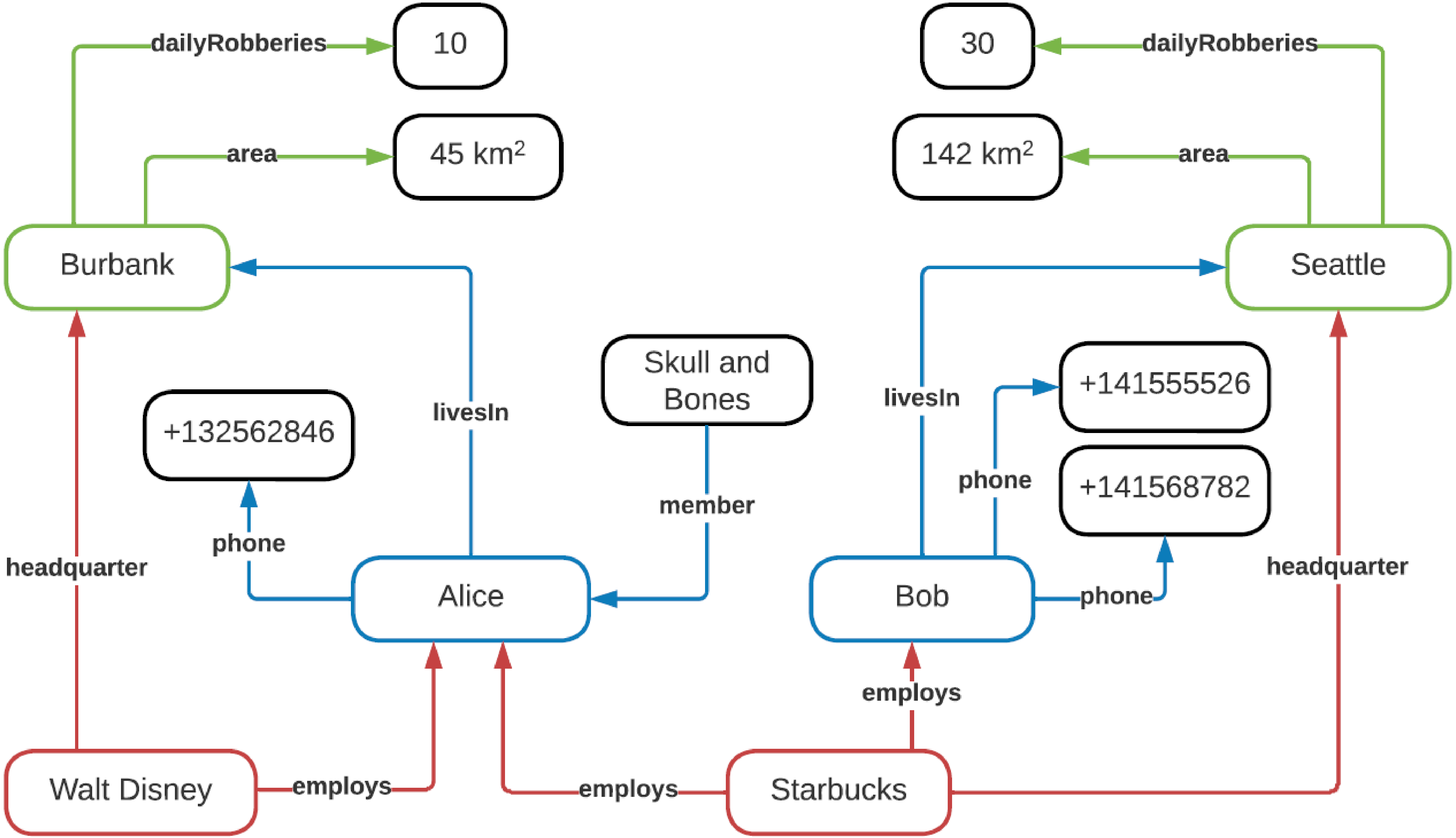
RDF (Resource Description Framework) Knowledge Graphs

Information about different *types* of **entities**



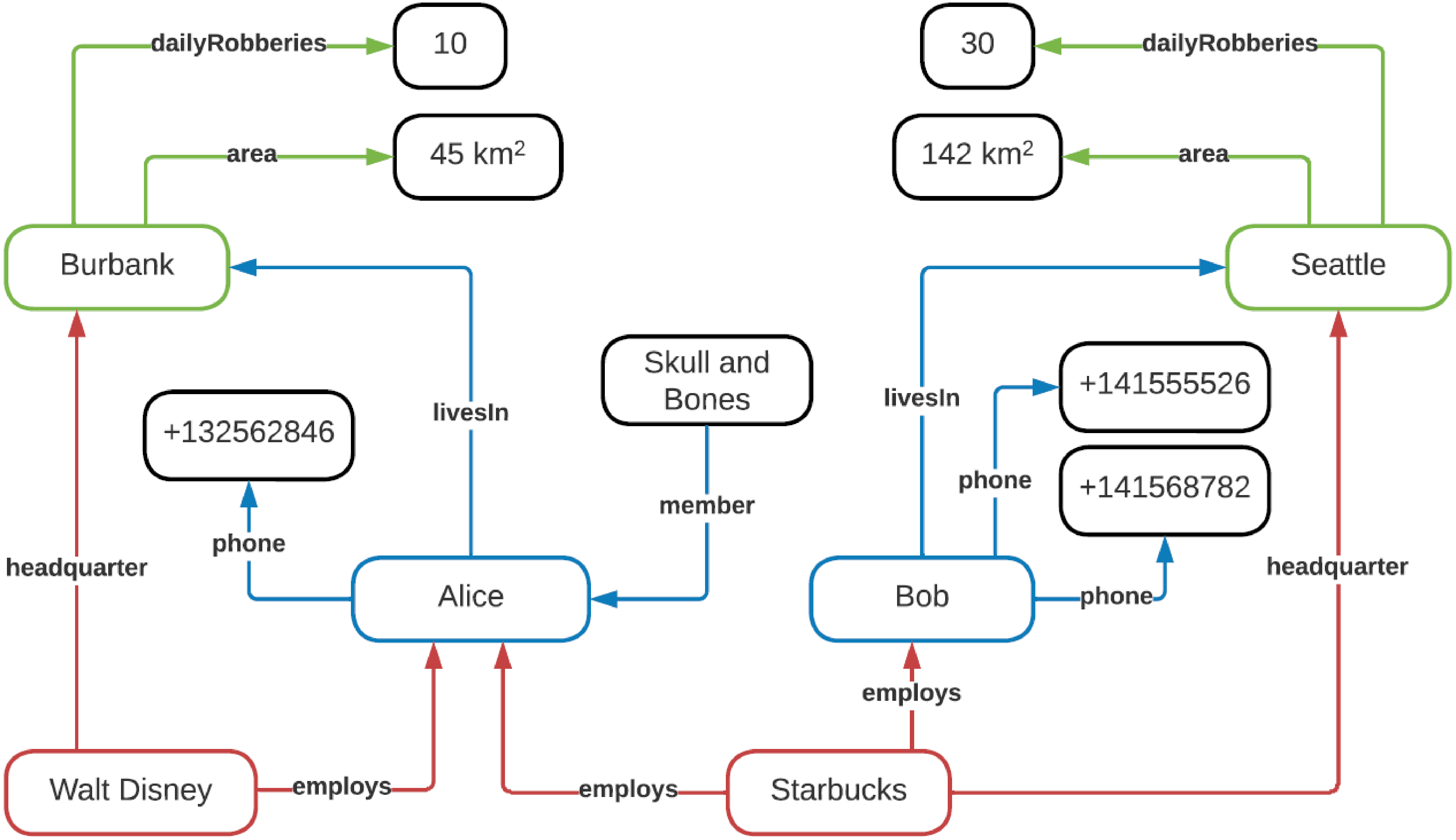
RDF (Resource Description Framework) **Knowledge Graphs**

Information about different types of entities -people



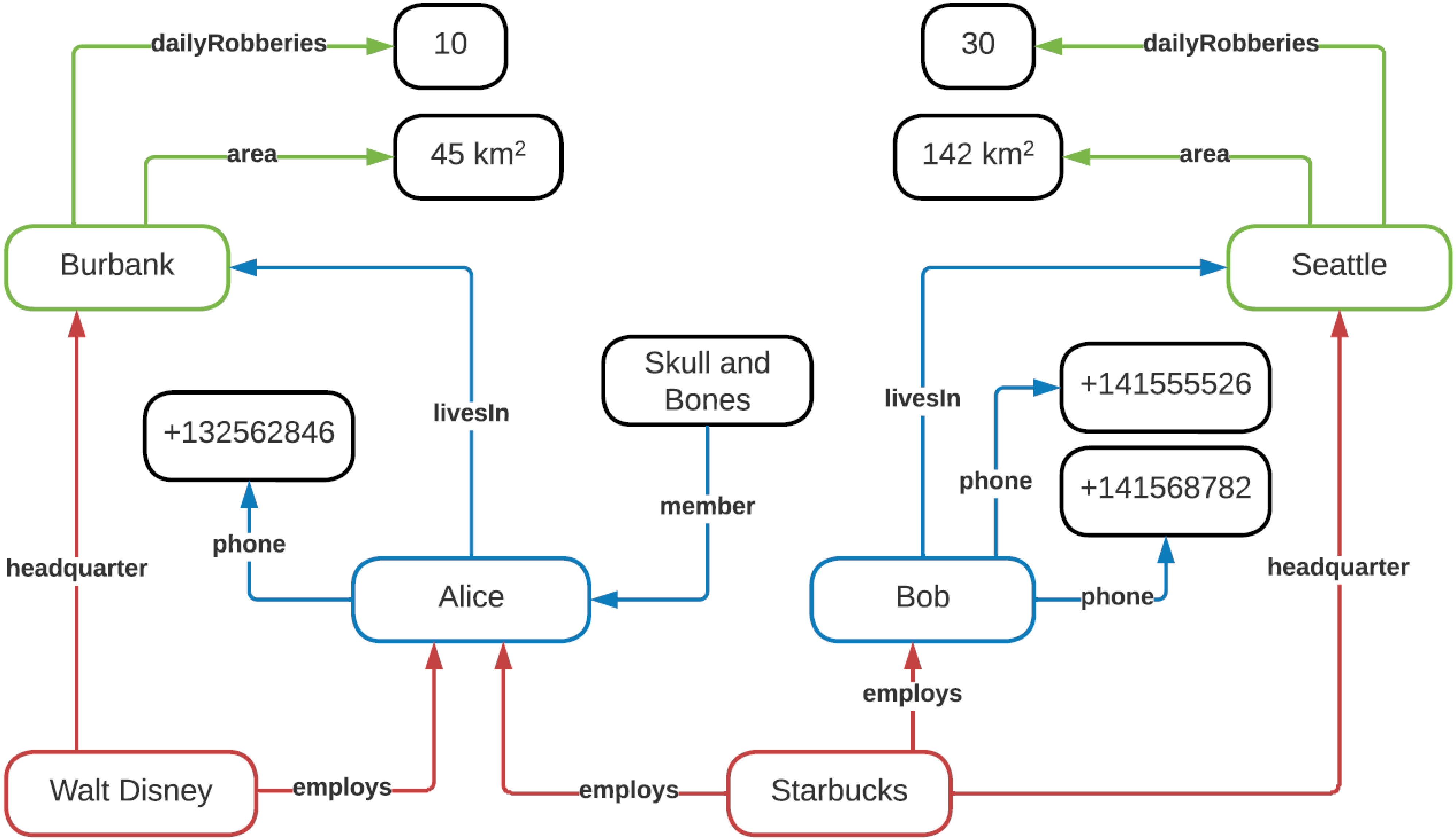
RDF (Resource Description Framework) Knowledge Graphs

Information about different types of entities
-people
-companies

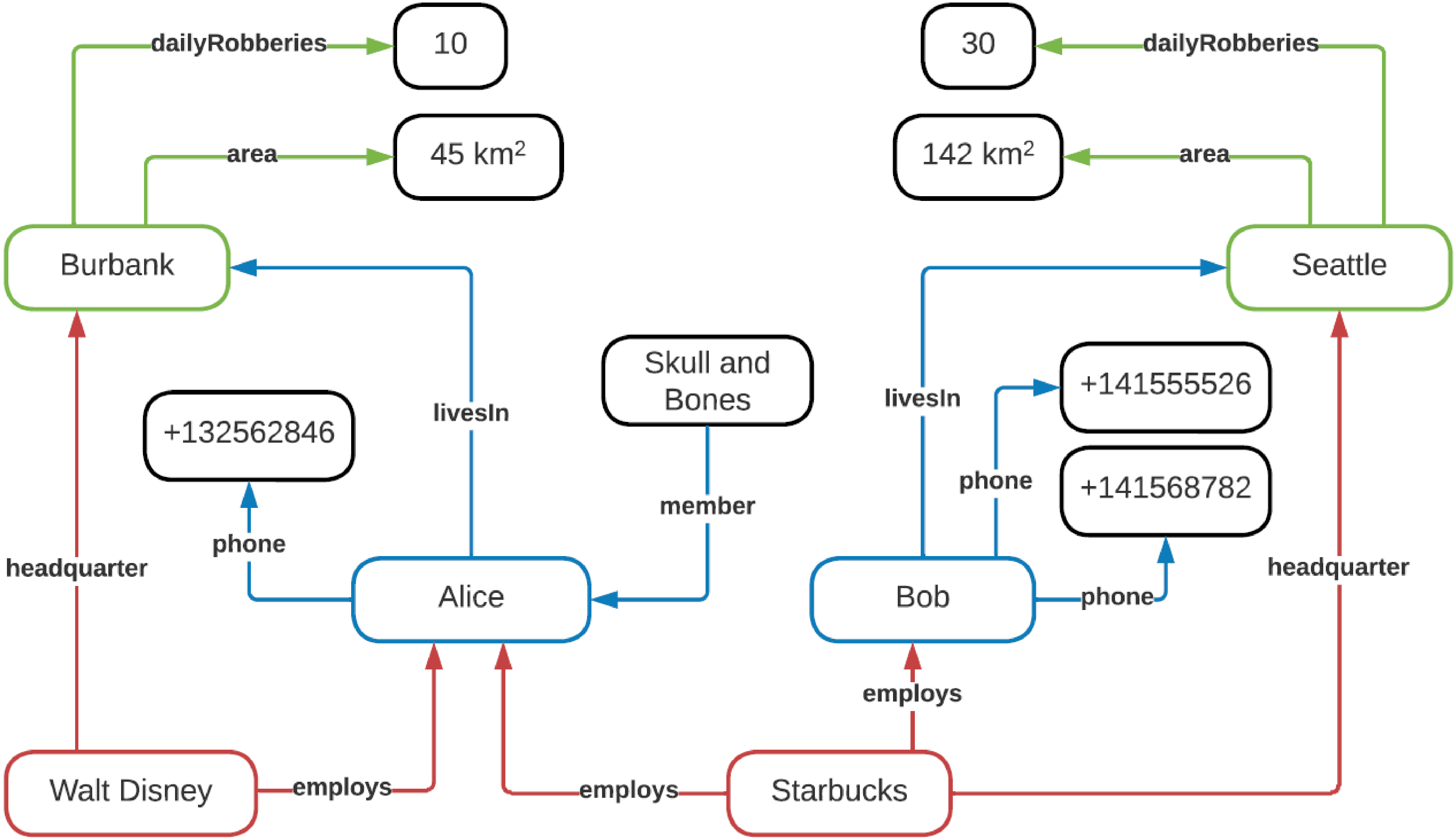


RDF (Resource Description Framework) Knowledge Graphs

Information about different types of entities
- people
- companies
- headquarter locations

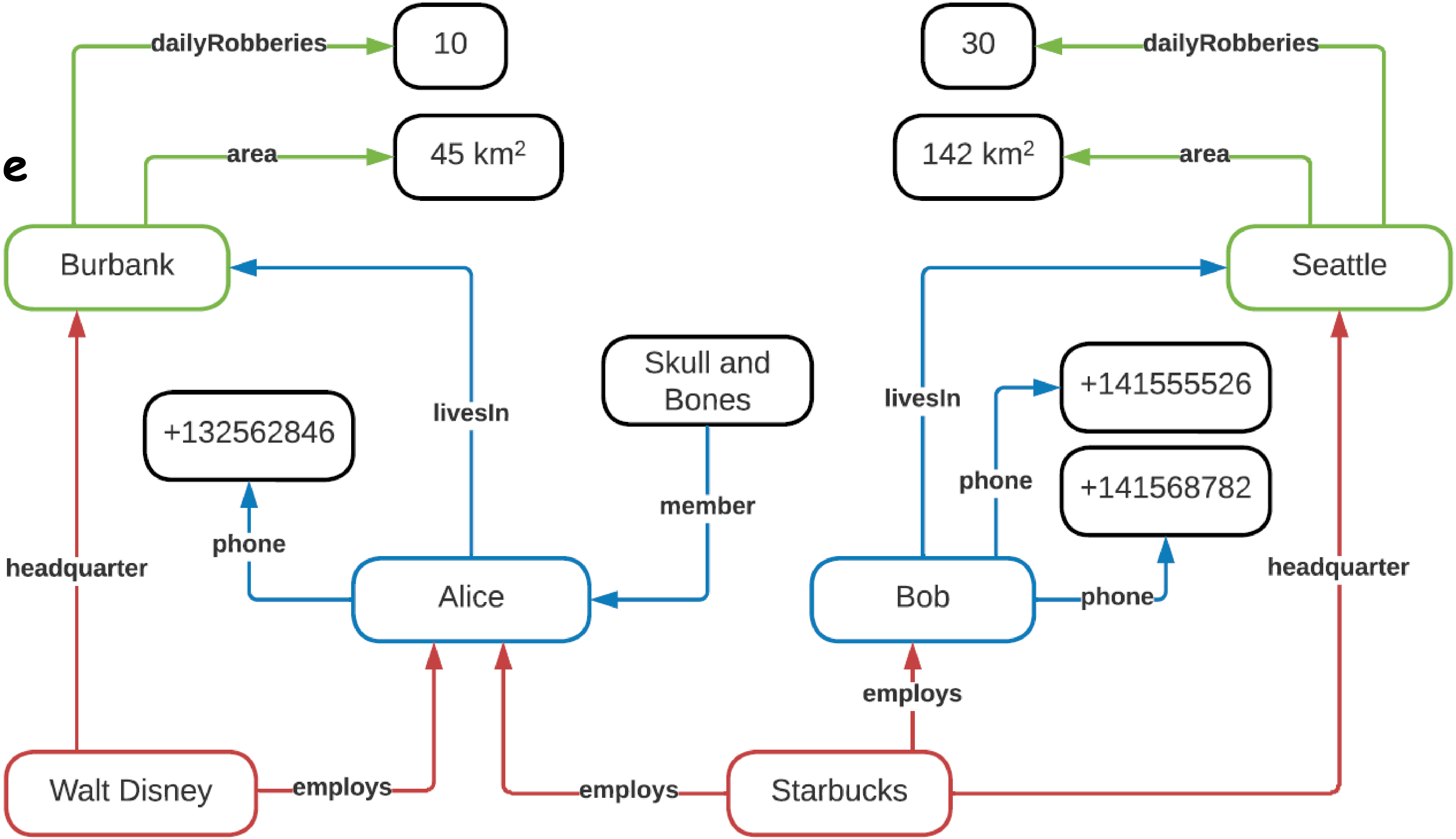


The goal: provide **aggregated properties** of the graph while protecting the privacy of **individual entities** within it



The goal: provide **aggregated properties** of the graph while protecting the privacy of **individual entities** within it

"**How many people** work in companies headquartered in a city with more than 20 daily robberies?"

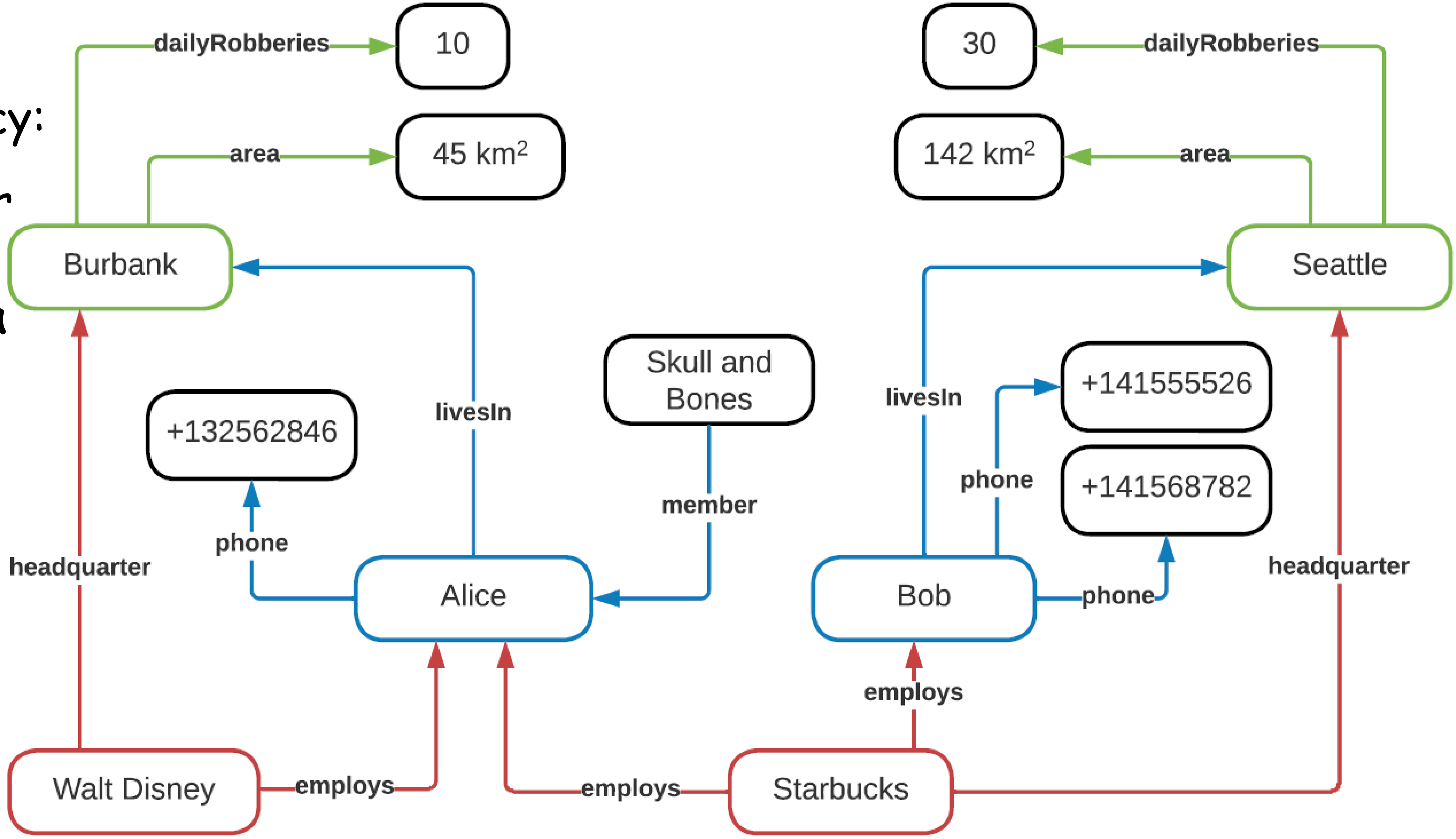


The goal: provide **aggregated properties** of the graph while protecting the privacy of **individual entities** within it

To support privacy:

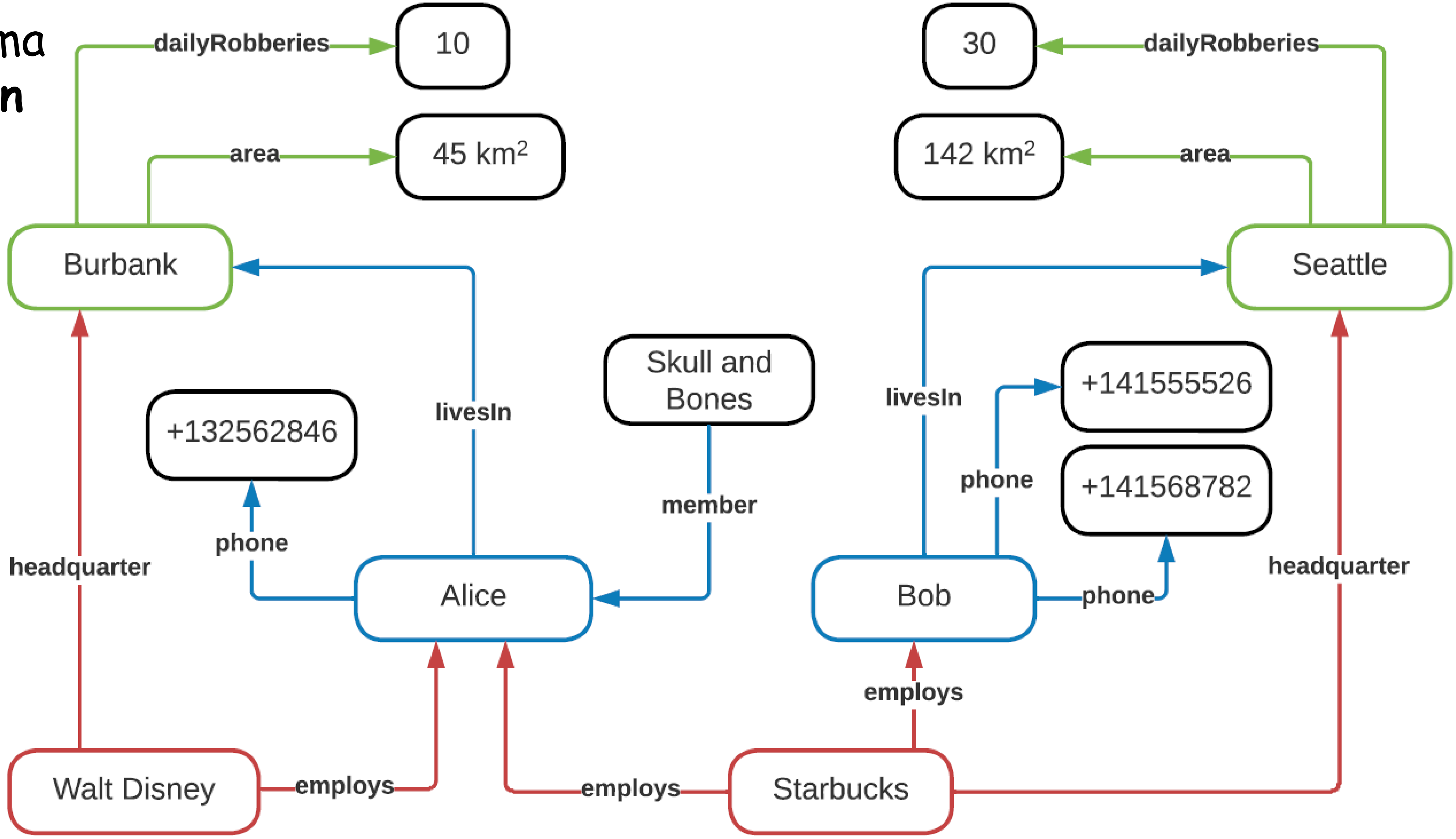
the administrator defines a **Privacy Schema** and identifies **entity types**

(this example has three types)



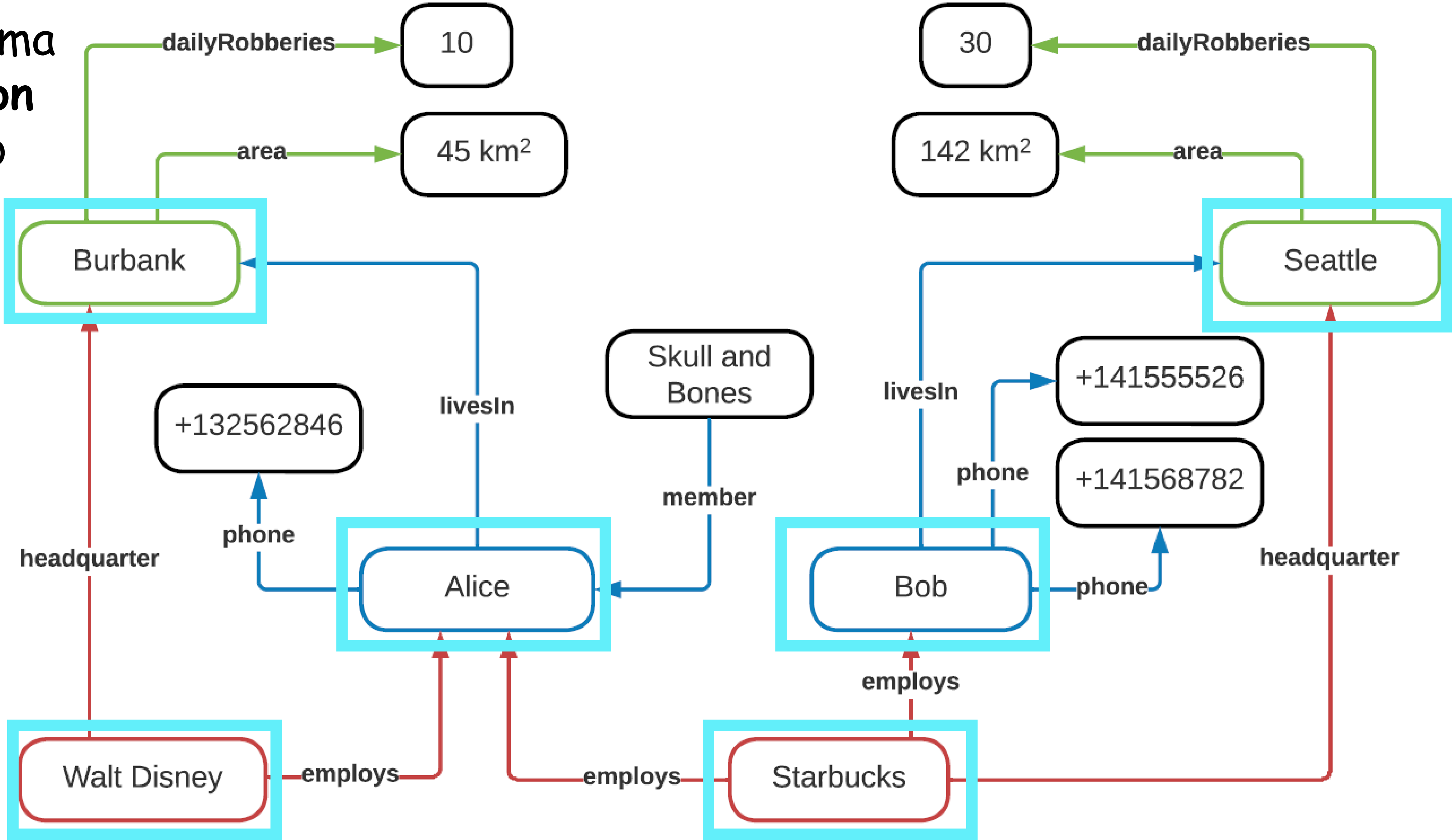
The goal: provide **aggregated properties** of the graph while protecting the privacy of **individual entities** within it

The Privacy Schema induces **a partition of the graph** into sub-graphs



The goal: provide **aggregated properties** of the graph while protecting the privacy of **individual entities** within it

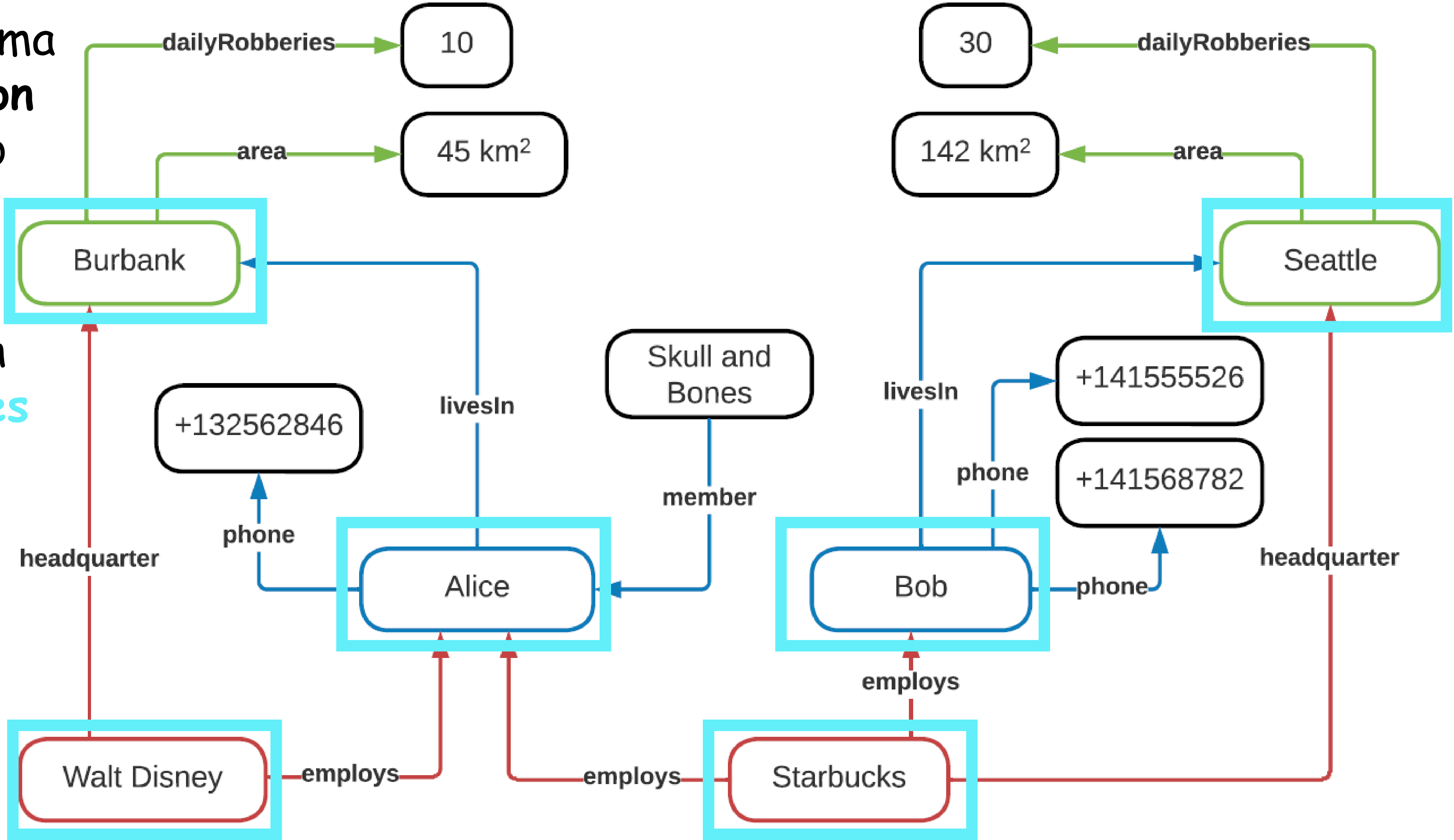
The Privacy Schema induces **a partition of the graph** into sub-graphs



The goal: provide **aggregated properties** of the graph while protecting the privacy of **individual entities** within it

The Privacy Schema induces **a partition of the graph** into sub-graphs

Each sub-graph is associated with **concreate entities** for privacy protection.

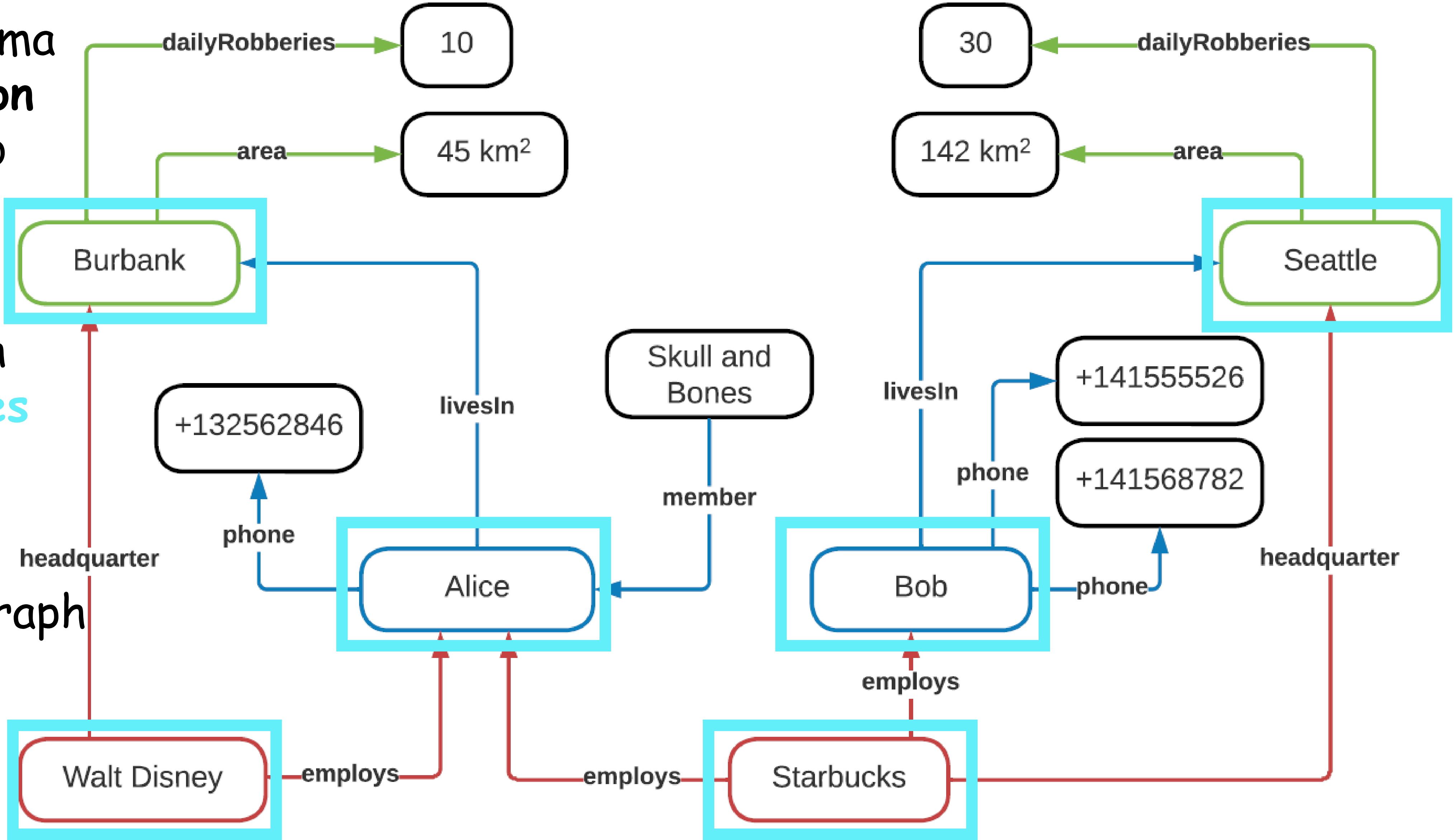


The goal: provide **aggregated properties** of the graph while protecting the privacy of **individual entities** within it

The Privacy Schema induces **a partition of the graph** into sub-graphs

Each sub-graph is associated with **concreate entities** for privacy protection.

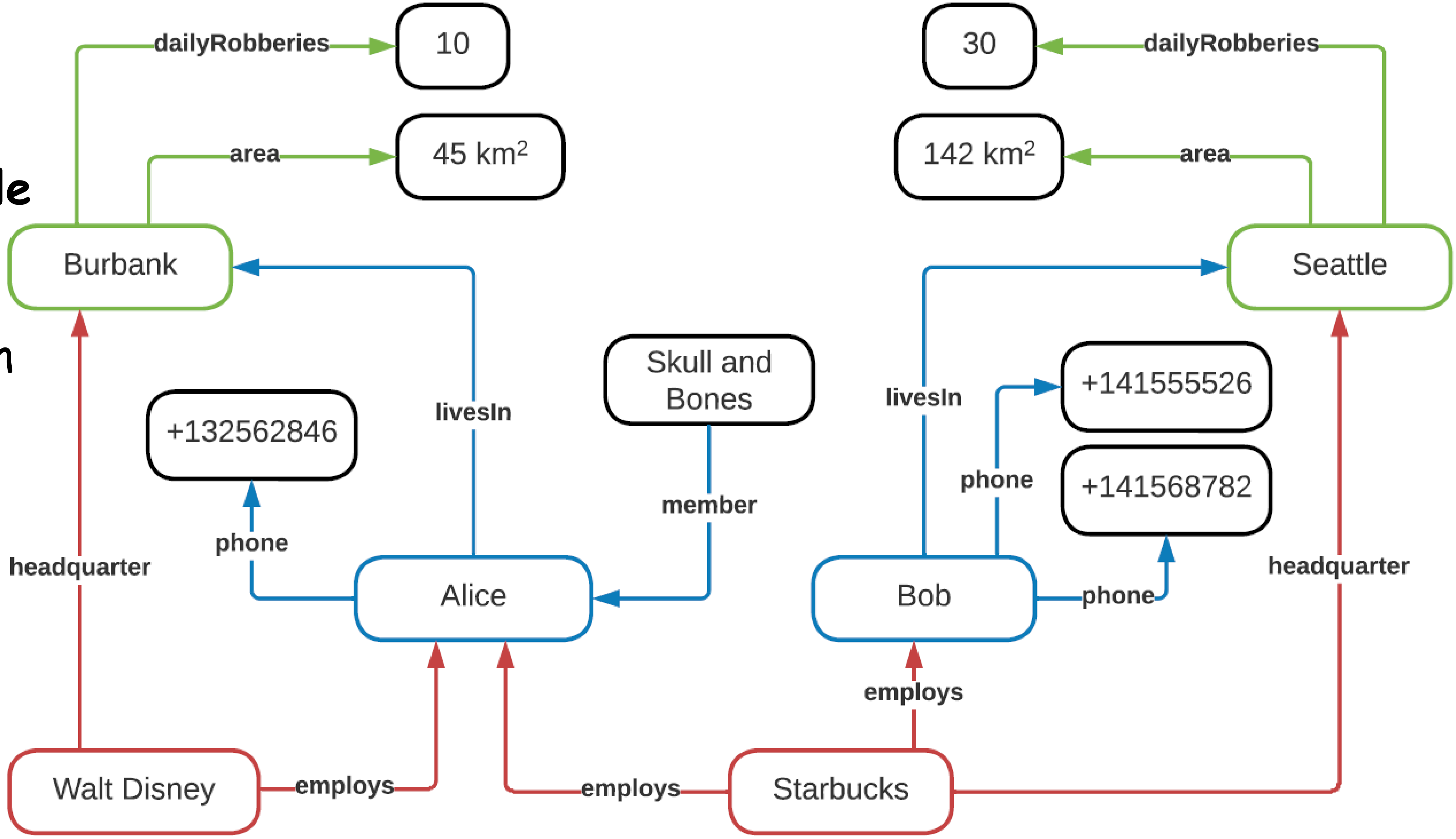
The **size** of the graph is the number of sub-graphs (this example is of size 6)



Breaking privacy by querying numerical properties of knowledge graphs

Reconstruction attacks:

"How many people work in a company headquartered in a city with more than 20 daily robberies?"



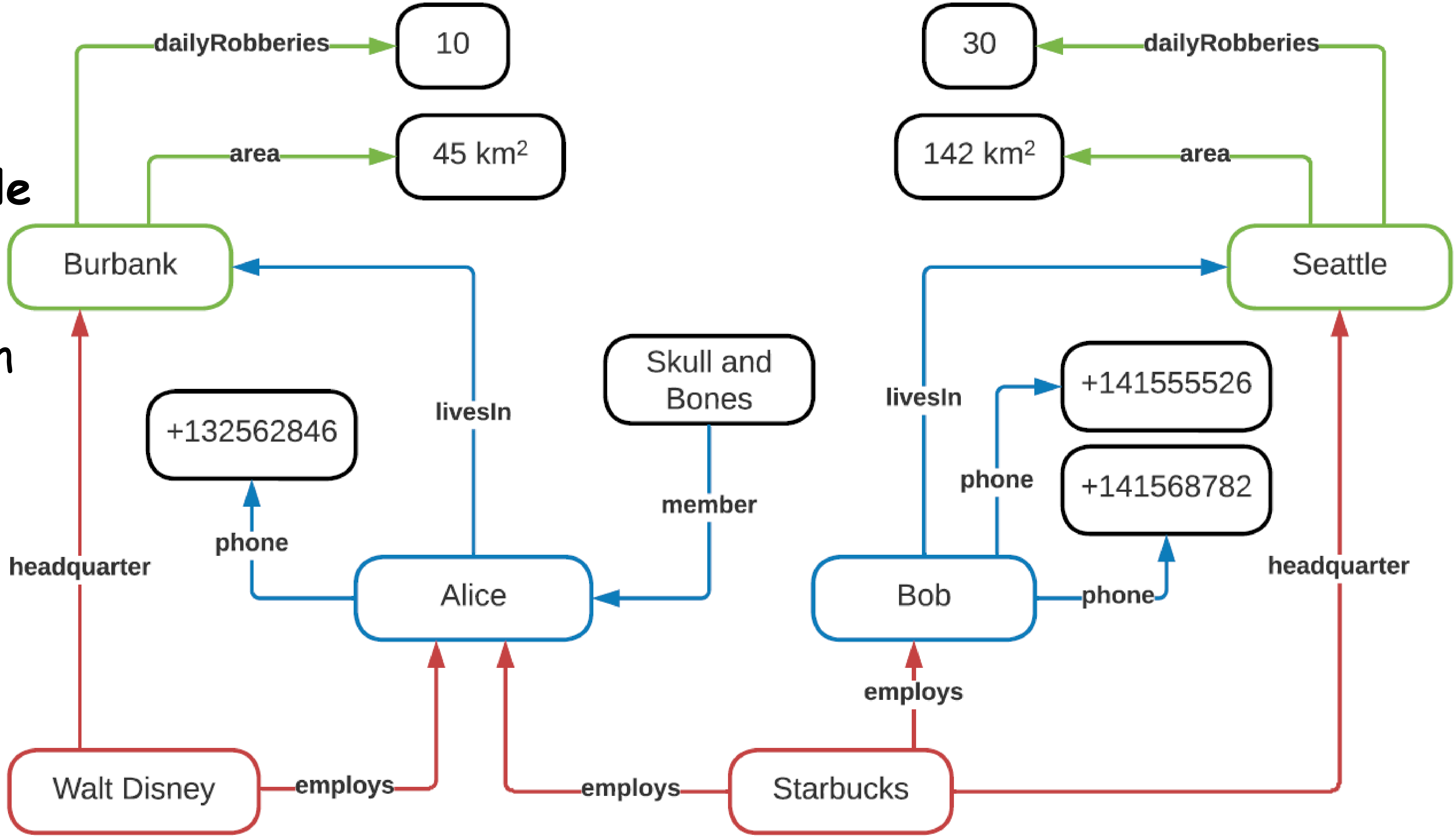
Breaking privacy by querying numerical properties of knowledge graphs

Reconstruction attacks:

"**How many people** work in a company headquartered in a city with more than 20 daily robberies?"

"**How many** ...in a city with more than 10 ...?"

⋮



Differential Privacy in terms of RDF Graphs

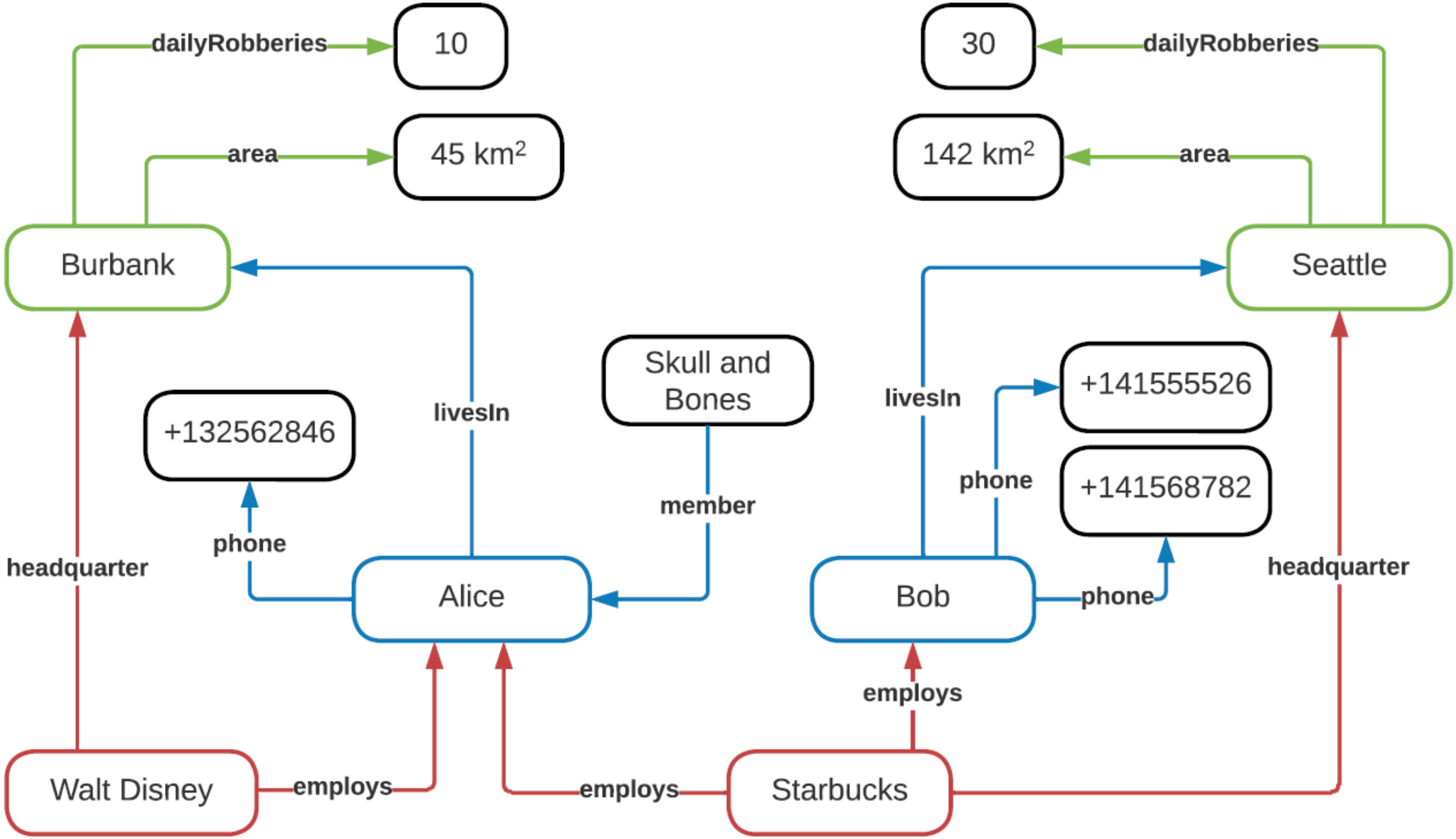
Informal:

- Given the set of real numbers R and the universe G of all possible graphs, a numerical query $f: G \rightarrow R$, is said to be differentially private if it yields indistinguishable results when applied to similar graphs g and g'.

SPARQL Numerical queries on knowledge graphs

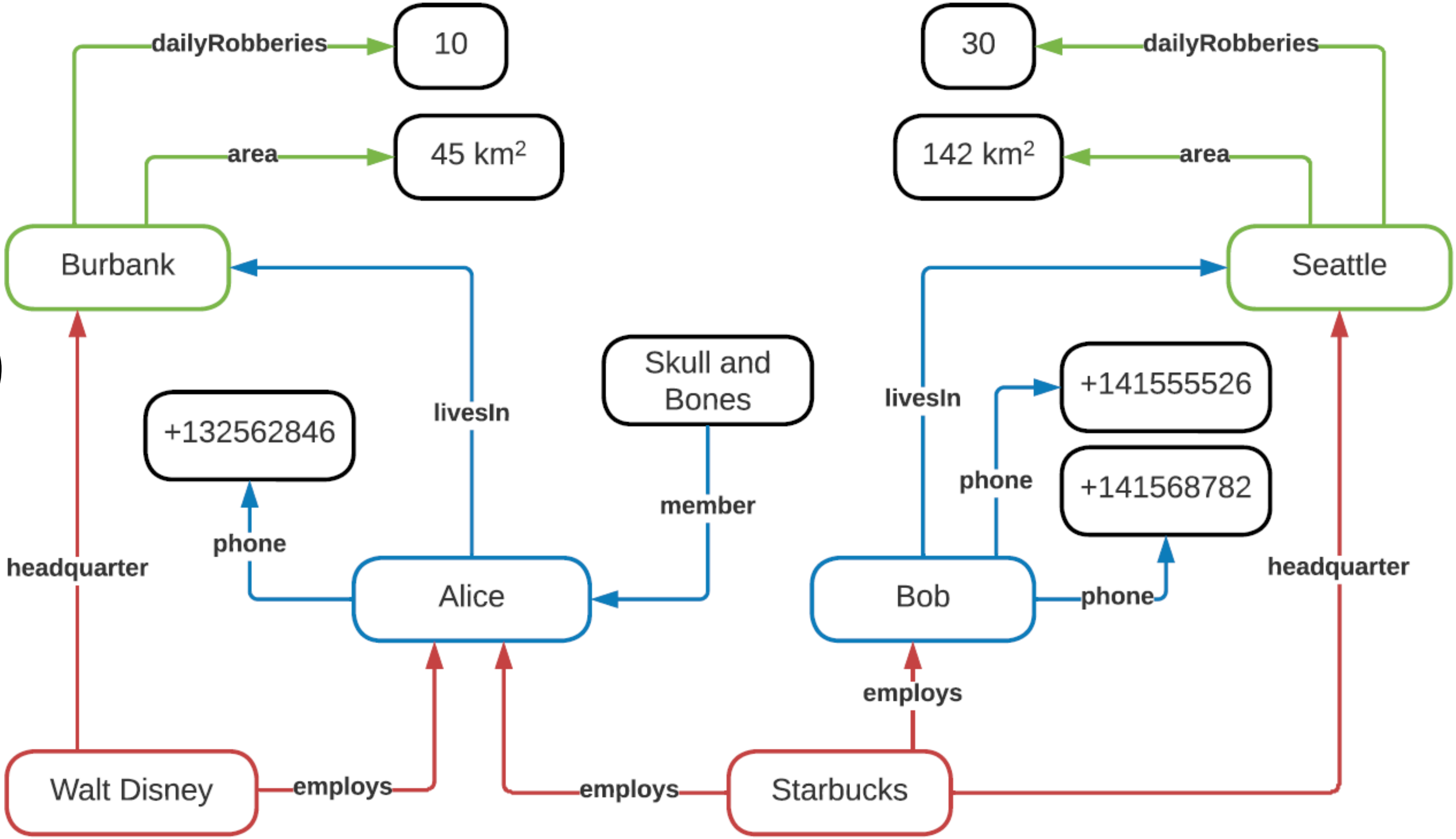
Three parts:

1. {(?c employs ?x),
   (?c headquarter ?h),
   (?h dailyRoberies ?n)
   }
2. {?n>20}
3. Count the number of different values of ?x



Similarities between knowledge graphs

A graph g' is at distance k of a graph g, d(g,g')=k, if g' can be obtained by changing (i.e. adding, deleting, or updating) subgraphs of g *associated* with k different entities.



Differential Privacy in terms of RDF Graphs

Formal:

- Graphs g and g' are neighbors (similar) if $d(g,g')=1$
- Let $\epsilon, \delta \geqslant 0$. A **randomized algorithm** A is

  (ϵ, δ)-differentially private

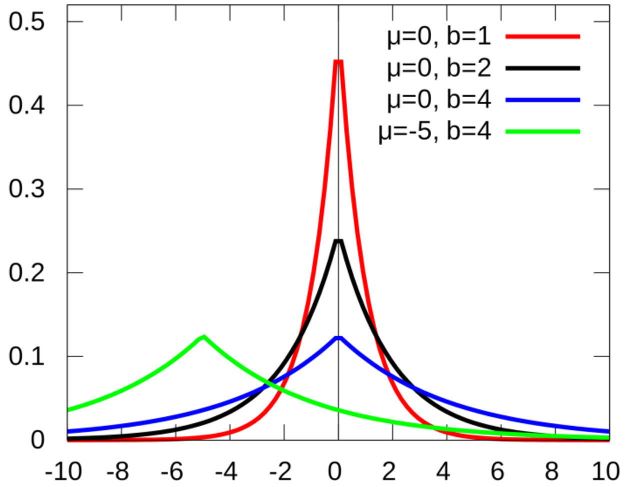
  if for every pair of neighboring graphs $g, g' \in G$ and every set $S \subseteq R$

  $$\Pr[A(g) \in S] \leqslant e^{\epsilon} \Pr[A(g') \in S] + \delta$$

The smaller the ϵ and δ, the closer these two probabilities are, and therefore, the less likely an adversary can tell $A(g)$ and $A(g')$ apart

Randomizing SPARQL Numerical Queries

- On input g, return f(g) plus some noise sampled from a Laplacian distribution:



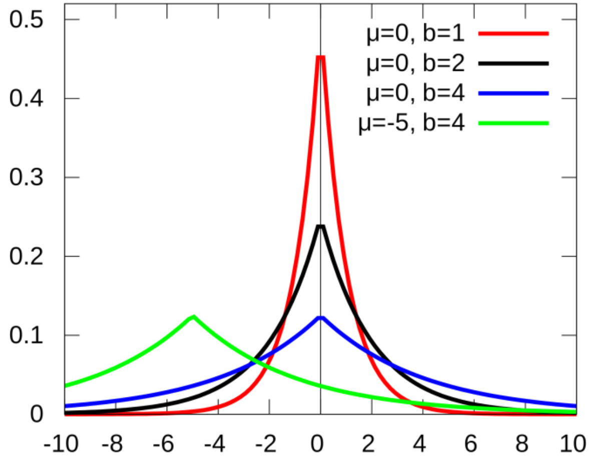
- Calibrate noise according to the local sensitivity LS_f of f with respect to a graph g, which measures f maximum variation upon neighboring graphs:

$$LS_f(g) = \max_{d(g,g')=1} | f(g) - f(g')|.$$

Theorem (see K. Nissim et al.)

Given a numeric query f: $G \to R$ of local sensitivity LS_f, and a smooth upper-bound U_f of LS_f, the randomized algorithm

$$A(g) = f(g) + Lap(U_f(g)/\epsilon)$$



is an (ϵ,δ)-differentially private version of f.

(δ is a parameter of the smoothing)

- Lap(b) represents a sample from the Laplacian distribution with pdf $\frac{1}{2b}e^{-|x|/b}$, mean 0 and variance $2b^2$.

Finding a smooth upper-bound of LS_f

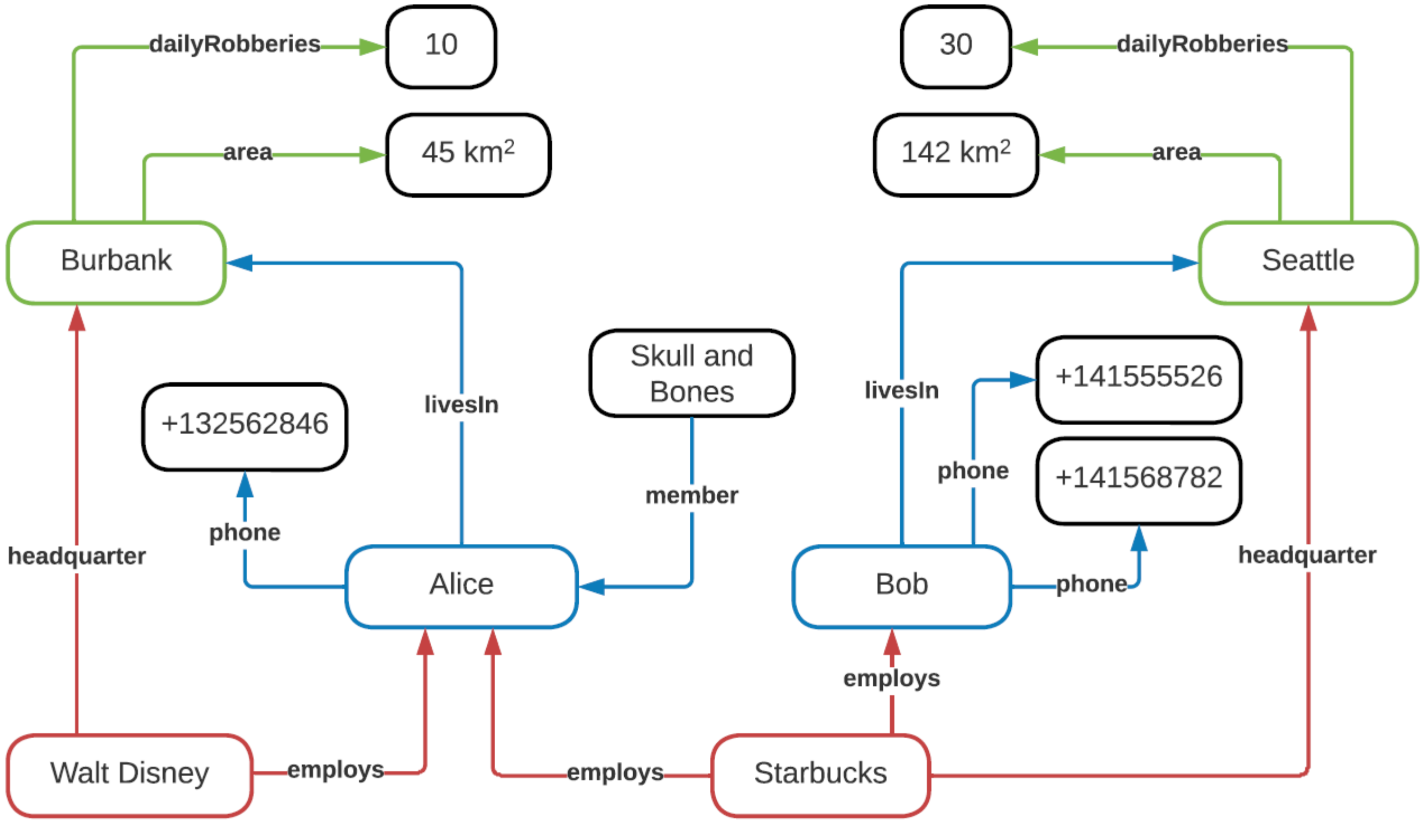
- For a query f: $G \rightarrow R$, we find a pointwise upper bound function of the local sensitivity of the query f at distance k, $U_f^{(k)}(g) \geqslant LS_f^{(k)}(g)$, for all $g \in G$, and then get

$$U_f(g) = max_{0 \leqslant k \leqslant size(g)}\ e^{-\beta k} U_f^{(k)}(g)$$

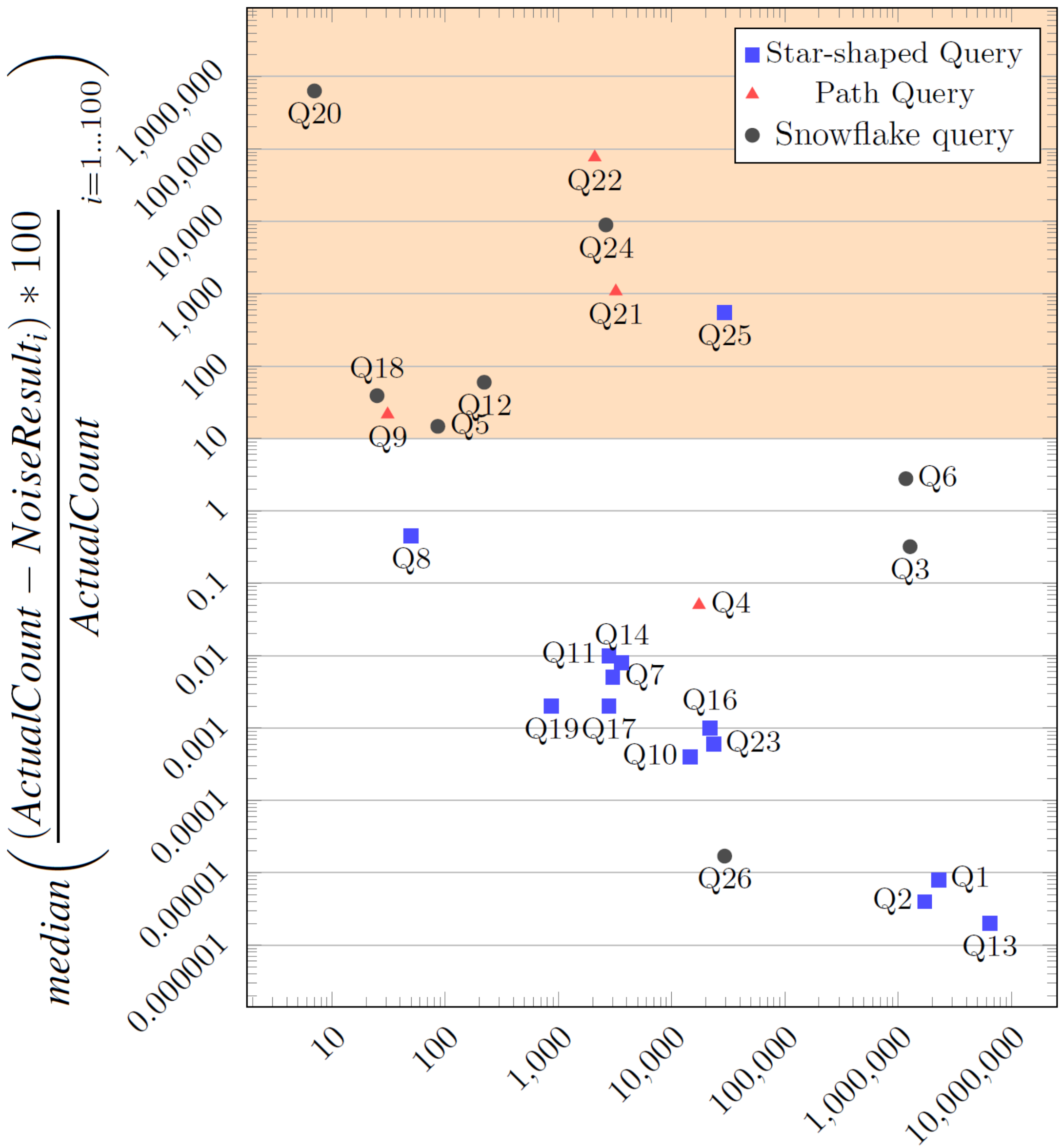
Which is a smooth upper bound of the local sensitivity $LS_f(g)$ of f on g.

The **most popular value** that can be assigned to the variables in f is used to calculate $U_f^{(k)}(g)$

(?c employs ?x),(?c headquarter ?h),(?h dailyRoberies ?n)



Evaluation Results



The End