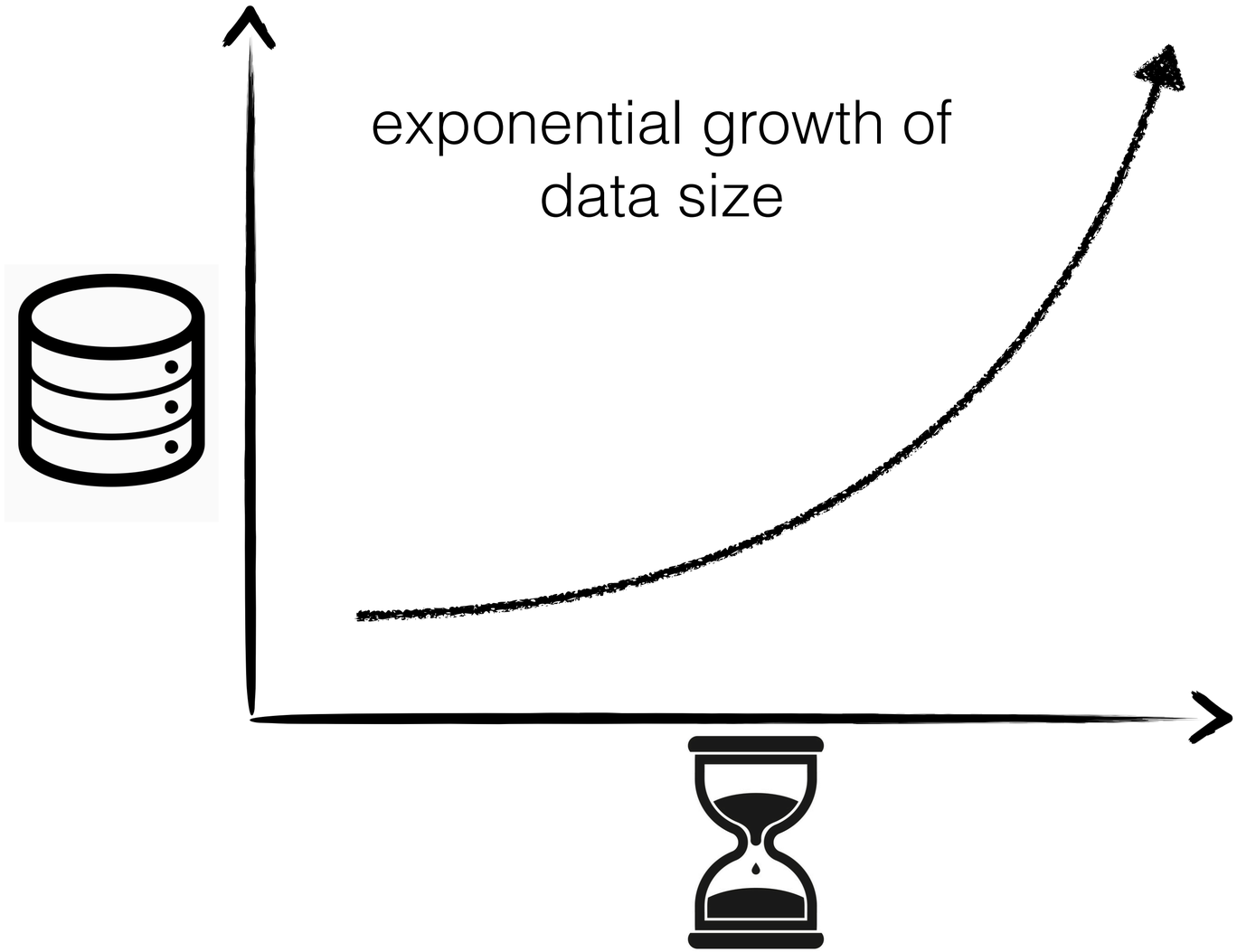
Query Language Support for Timely Data Deletion

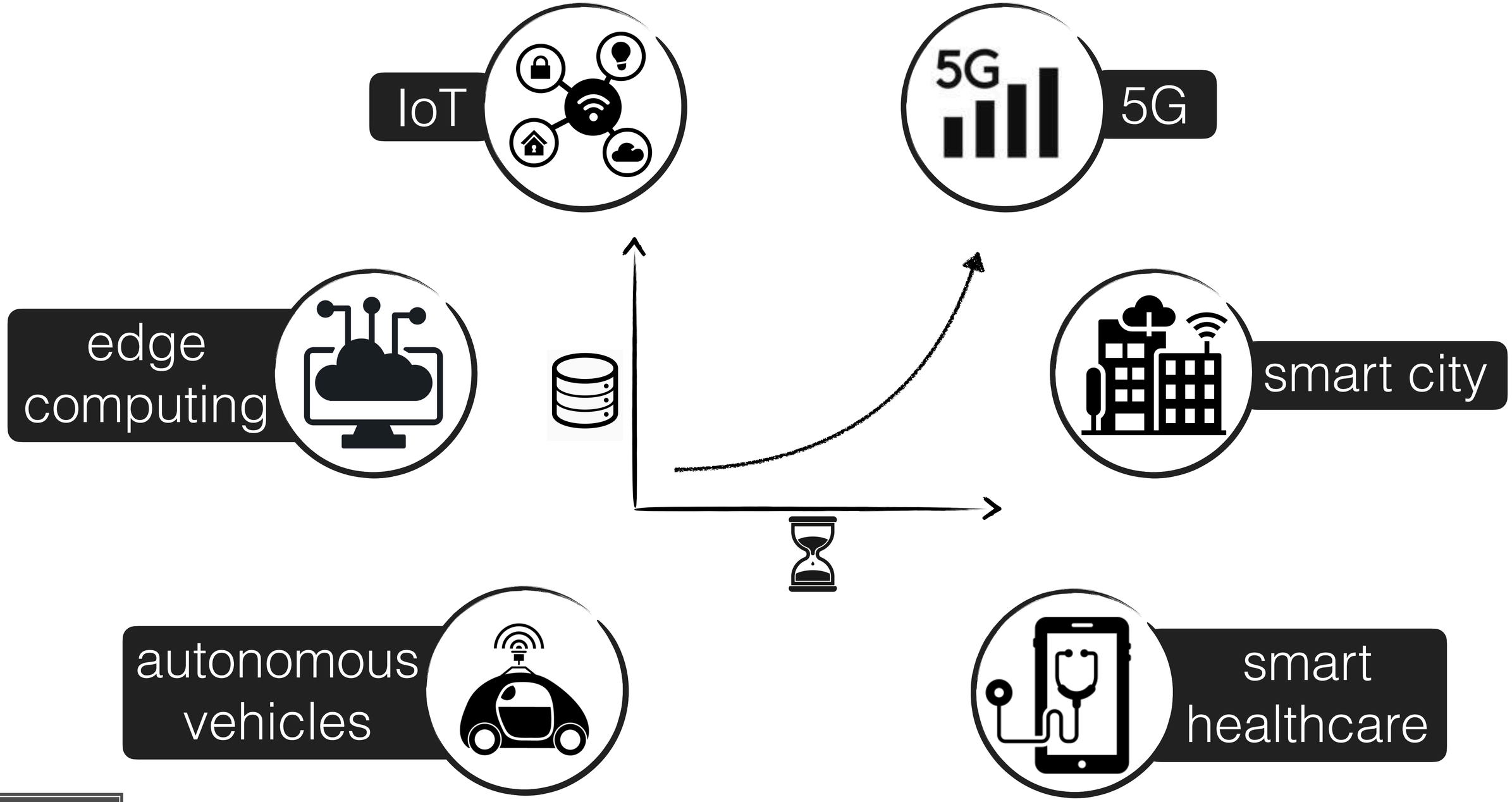
Subhadeep Sarkar

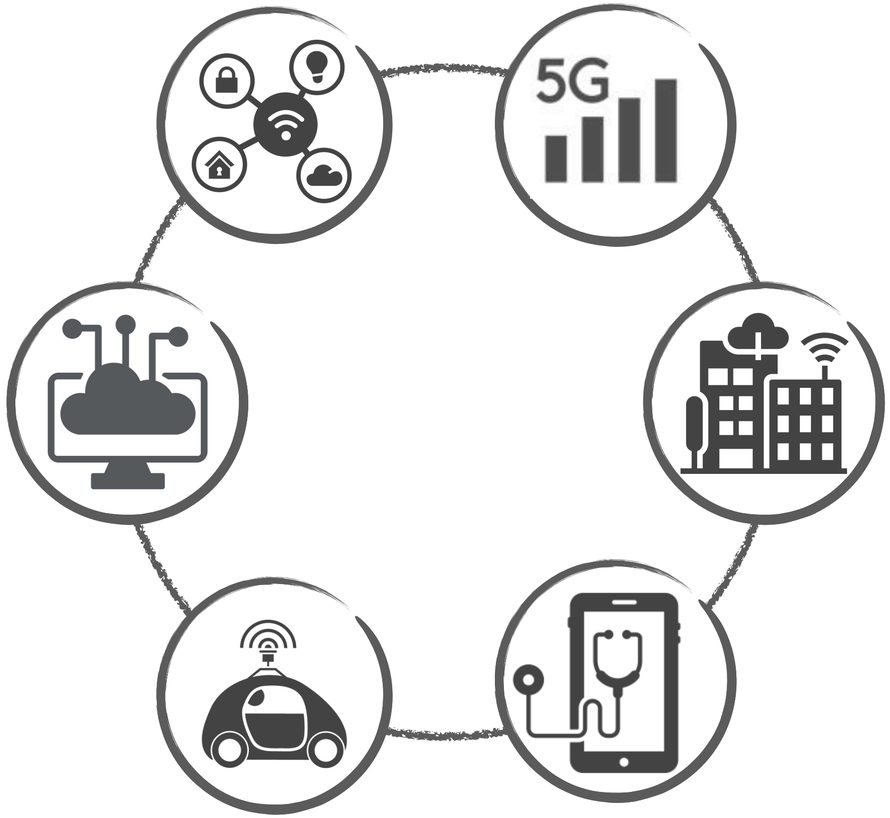
Manos Athanassoulis

BOSTON UNIVERSITY

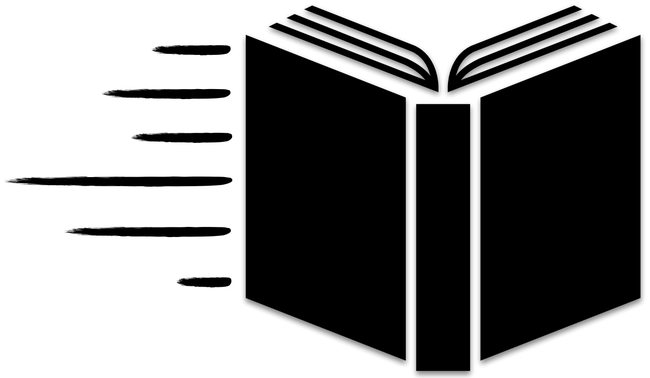
DiSC lab



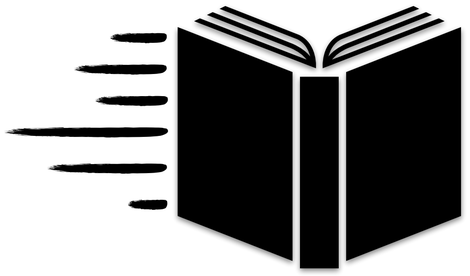

IoT
5G
5G
edge computing
smart city
autonomous vehicles
smart healthcare



fast writes



fast reads



Out-of-place paradigm

BOSTON UNIVERSITY

Out-of-place **systems**

Relational & Array-based

NoSQL

Bigtable

levelDB

QuasarDB

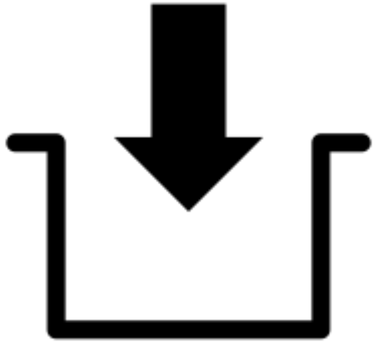
Out-of-place **systems**

Relational & Array-based

NoSQL

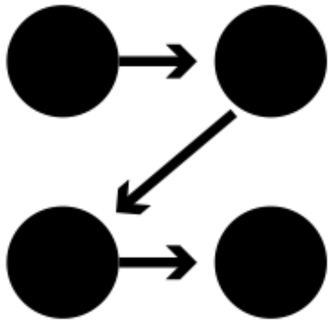
Bigtable

Out-of-place paradigm



Realizes
updates & deletes
through new inserts

Applies
updates/deletes
lazily to base data



Maintains
ingestion-order
of entries

BOSTON
UNIVERSITY

Deletes in out-of-place systems

ID	Flag	Name	Dept	Salary
5	0	John	CS	40000
12	0	Joe	Math	35000
3	0	Amar	Math	30000
7	0	Saka	CS	30000
21	0	Li	Math	60000

```
DELETE FROM table
WHERE ID = 7;
```

Deletes in out-of-place systems

ID	Flag	Name	Dept	Salary
7	**1**	**-**	**-**	**-**
5	0	John	CS	40000
12	0	Joe	Math	35000
3	0	Amar	Math	30000
7	0	Saka	CS	30000
21	0	Li	Math	60000

```
DELETE FROM table
WHERE ID = 7;
```

```
INSERT INTO table (ID, Flag)
VALUES (7, 1);
```

BOSTON UNIVERSITY

Deletes in out-of-place systems

newer

ID	Flag	Name	Dept	Salary
7	**1**	**-**	**-**	**-**
5	0	John	CS	40000
12	0	Joe	Math	35000
3	0	Amar	Math	30000
7	0	Saka	CS	30000
21	0	Li	Math	60000

older

```
DELETE FROM table
WHERE ID = 7;
```

```
INSERT INTO table (ID, Flag)
VALUES (7, 1);
```

BOSTON UNIVERSITY

Deletes in out-of-place systems

ID	Flag	Name	Dept	Salary
7	**1**	**-**	**-**	**-**
5	0	John	CS	40000
12	0	Joe	Math	35000
3	0	Amar	Math	30000
7	0	Saka	CS	30000
21	0	Li	Math	60000

```
DELETE FROM table
WHERE ID = 7;
```

```
INSERT INTO table (ID, Flag)
VALUES (7, 1);
```

BOSTON UNIVERSITY

GDPR
(EU, UK)

CCPA, CPRA
(California)

VCDPA
(Virginia)

Right to be forgotten

Right to delete

Deletion right

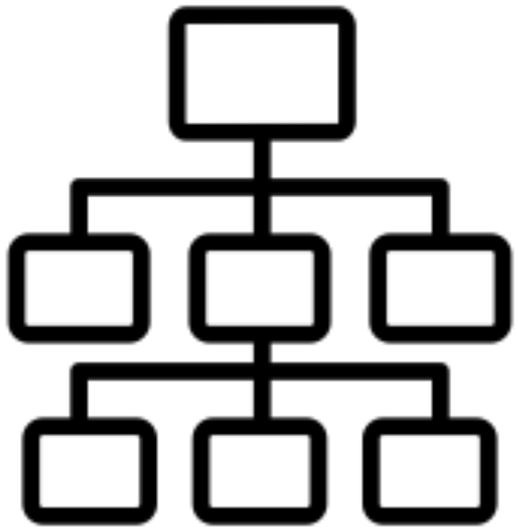
Policy layer

Right to be forgotten

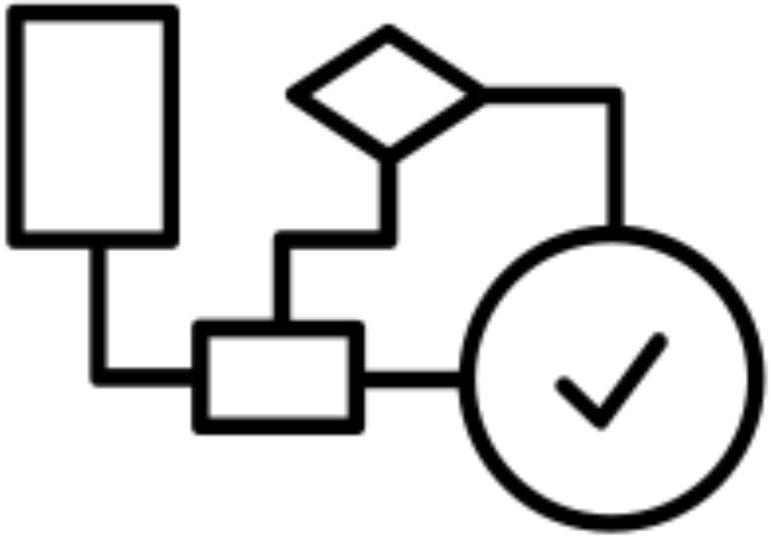
Right to delete

Deletion right

System layer



Data layout re-organization



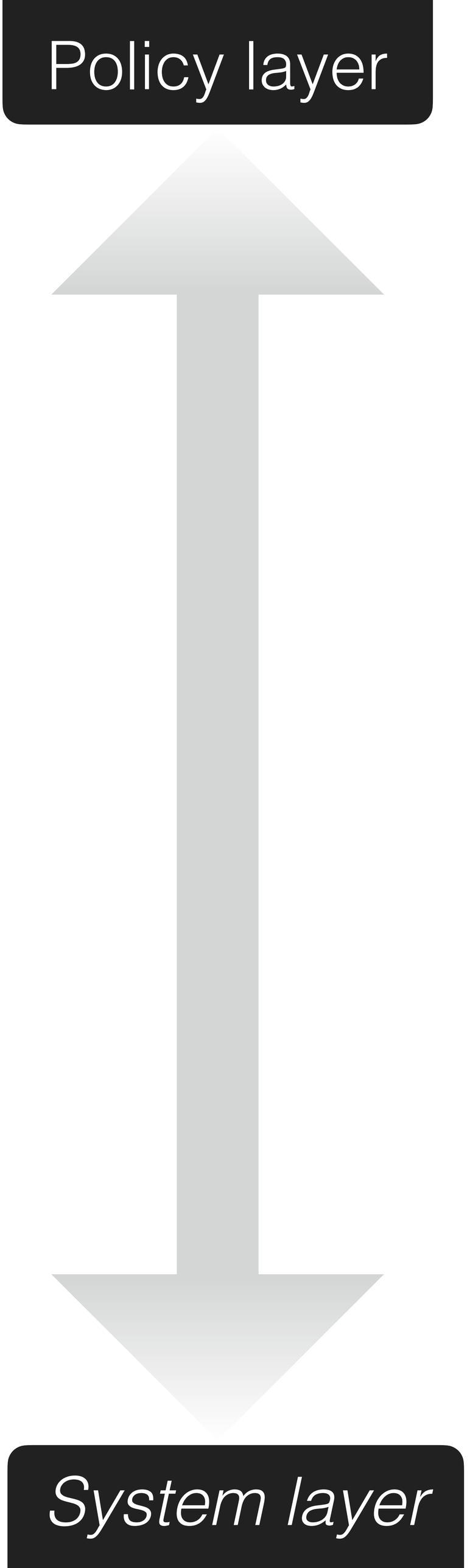
Data deletion algorithms

Sarkar et al., SIGMOD ’20

Cohn-Gorrdon et al., USENIX Security Symp. ’20

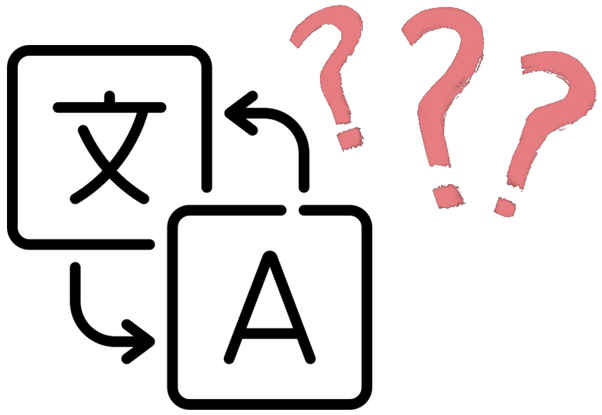
BOSTON UNIVERSITY

BOSTON UNIVERSITY



Policy layer

deletion requirements



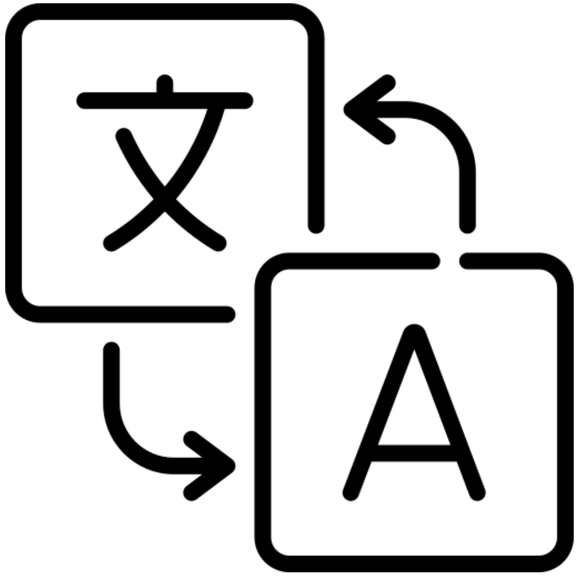
System layer

```
50
51  int main(int argc, char *argv[]) {
52    // check emu_environment.h for the contents of EmuEnv
53    EmuEnv* _env = EmuEnv::getInstance();
54    Stats* fade_stats = Stats::getInstance();
55    //parse the command line arguments
56    if (parse_arguments2(argc, argv, _env)){
57      exit(1);
58    }
59
60    if (_env->verbosity >= 4) {
61      std::cout << "printing del_per_lat" << std::endl;
62      for (int i = 0; i < 2; ++i){
63        std::cout << i << ": " << EmuEnv::GetLevelDeletePe
64      }
65    }
66
67    int s = runWorkload(_env);
68
69    fade_stats->printStats();
70
71    return 0;
72  }
73
74
75  void configOptions(EmuEnv* _env, Options *op, BlockBasedTableOptions
76      // *op = Options();
```

BOSTON UNIVERSITY

Policy layer

deletion
requirements



requirement
translation

System layer

BOSTON
UNIVERSITY

Deletion **requirements**

retention-based

delete ***all data older than D days***

on-demand

delete ***data object X within D days***

BOSTON UNIVERSITY

Retention-based deletes

```
CREATE TABLE R (ID int, Name varchar(255), ...)
  WITH FIXED RET_DUR (t1 180, t2 365);

INSERT INTO R (32, Aaron, ...)
  WHERE RET_DUR = t1;
```

Retention-based deletes

```
CREATE TABLE R (ID int, Name varchar(255), ...)
    WITH FIXED RET_DUR (t1 180, t2 365);
```

```
INSERT INTO R (32, Aaron, ...)
    WHERE RET_DUR = t1;
```

BOSTON UNIVERSITY

Retention-based deletes

```
CREATE TABLE R (ID int, Name varchar(255), ...)
    WITH ARBITRARY RET_DUR;

INSERT INTO R (32, Aaron, ...)
    WITH RET_DUR 90;
```

On-demand deletes

```
CREATE TABLE R (ID int, Name varchar(255), ...)
    WITH FIXED DPT (d1 30, d2 45, d3 60);

DELETE FROM R
    WHERE ID = 32
        WITH DPT d2;
```

SQL-support for deletes

```
CREATE TABLE R (column1 type1, column2 type2, ...)
    WITH RET_DUR
        {ARBITRARY|FIXED (t1 <ret1>, t1 <ret1>, ...)}
    WITH DPT
        {ARBITRARY|FIXED (d1 <dpt1>, d1 <dpt1>, ...)};
```

```
INSERT INTO R (val1, val2, ...)
    WITH RET_DUR {<t>|t<i>};
```

```
DELETE FROM R
    WHERE (...)
        WITH DPT {<d>|d<i>};
```

BOSTON UNIVERSITY

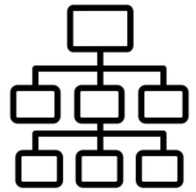
Policy layer

Right to be forgotten

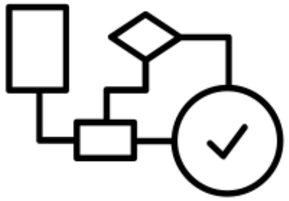
Right to delete

Deletion right

System layer



Data layout re-organization



Data deletion algorithms

BOSTON UNIVERSITY

Policy layer

Right to be forgotten

Right to delete

Deletion right

Requirements layer

Retention-based Deletes

On-demand Deletes

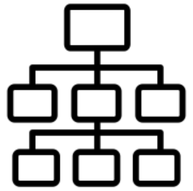
Application layer

```
CREATE TABLE R (...)
  WITH RET_DUR
  {ARBITRARY | FIXED(...)}
  WITH DPT
  {ARBITRARY | FIXED(...)};
```

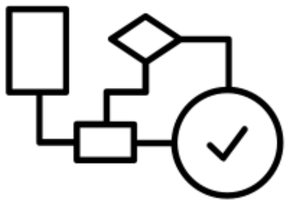
```
INSERT INTO R (...)
  WITH RET_DUR {<t>|t<i>};
```

```
DELETE FROM R
  WHERE (...)
    WITH DPT {<d>|d<i>};
```

System layer



Data layout re-organization

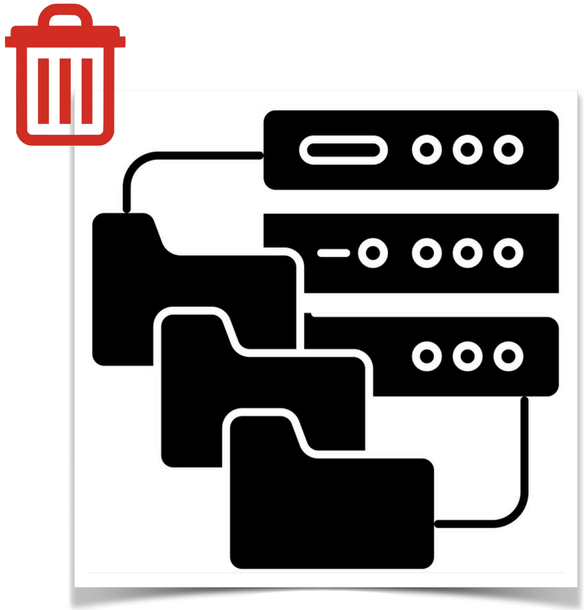


Data deletion algorithms

BOSTON UNIVERSITY

System layer

APIs to express system deletion requirements



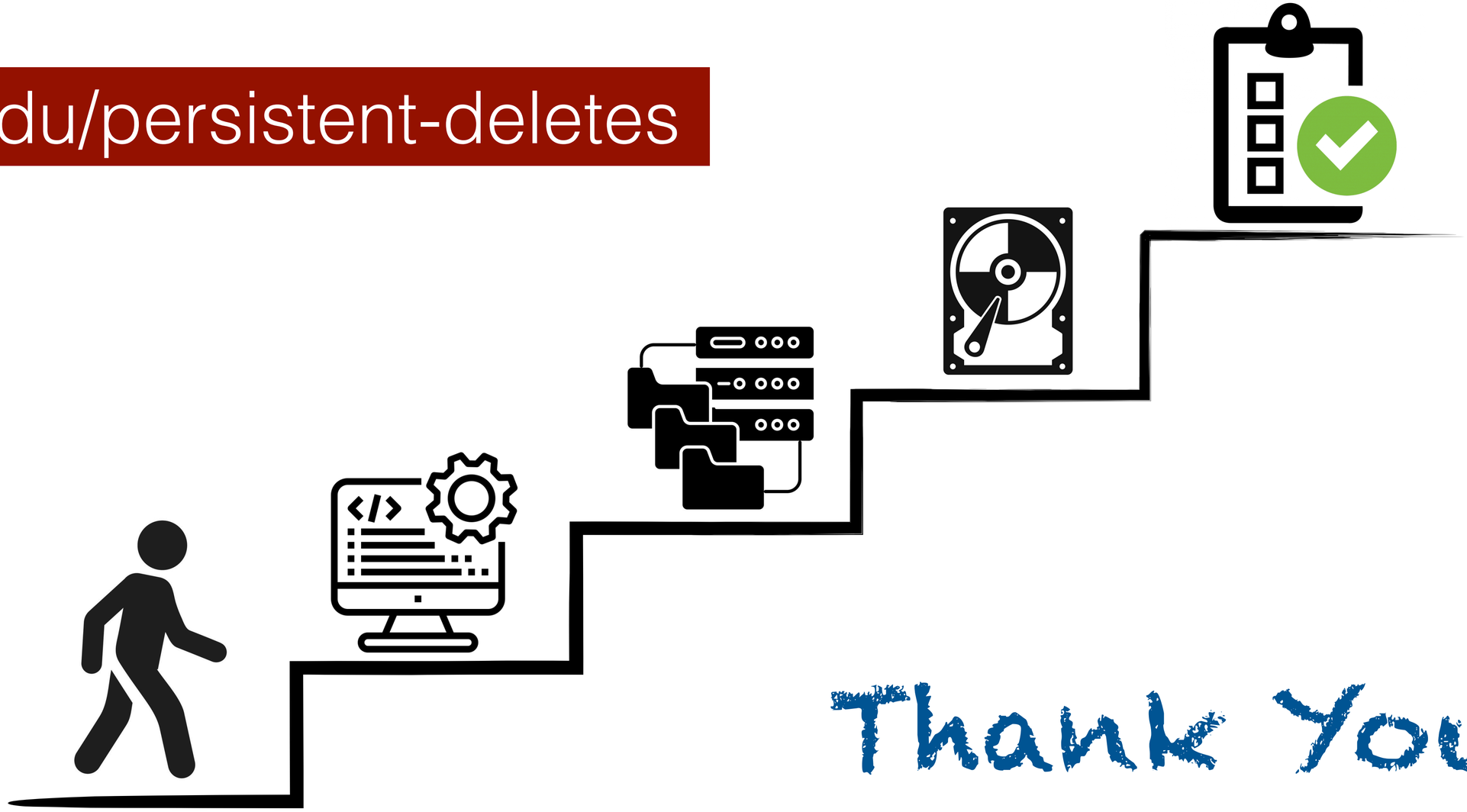
Persistent deletion from file systems

Secure & persistent deletion at device level

BOSTON UNIVERSITY

Toward **deletion-compliant** data systems

disc.bu.edu/persistent-deletes



Thank You!

BOSTON UNIVERSITY