SmartSPEC: Customizable Smart Space Datasets via Event-Driven Simulations

Andrew Chio[1], Daokun Jiang[1], Peeyush Gupta[1], Georgios Bouloukakis[2], Roberto Yus[3], Sharad Mehrotra[1], Nalini Venkatasubramanian[1]

Virtual, March 23, 2022

IEEE PerCom 2022

[1] University of California, Irvine

[2] Télécom SudParis, IP Paris

[3] University of Maryland, Baltimore County

IoT-Enabled Smart Spaces

Internet-of-Things (IoT)

Healthcare


Credit: *Emergency Sign*, *CC0*

Hospitals


Credit: *Senior Living*, *by Leisure Care*, *CC-BY-SA*

Senior Homes

Facility Management


Credit: *Clear glass pump bottle on ceramic sink*, *CC0*

Consumable Monitoring


Credit: Elevator lobby, Appraisers Building, San Francisco, California, *PDM*

Elevators

Safety


Credit: CCTV, *CC0*

Stores


Credit: Smoke detector 01, *CC-BY-SA*

Residential Homes

• • •

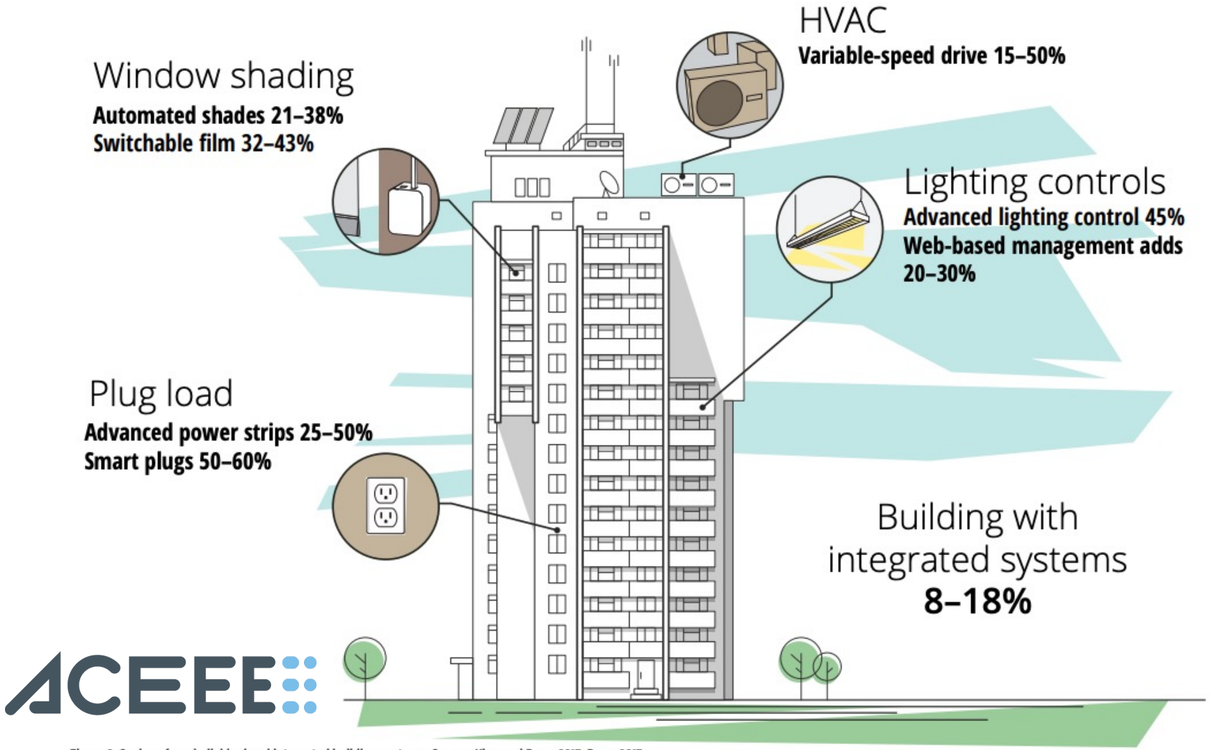


Figure 2. Savings from individual and integrated building systems. *Source:* King and Perry 2017; Perry 2017.

Benefits:

- Energy Efficiency, Sustainability
- Building Resilience, Reliability
- Adaptability to Dynamic Conditions

Towards Smarter Buildings: The Need for Realistic Data

Heterogeneity, Scalability, Portability, Robustness

Fire Evacuation in a High-Rise Building

- **Realistic data is necessary to test and validate smart space approaches in heterogeneous human environments**
  - Evaluating robustness of algorithms
  - Failure testing
  - Scalability testing
  - Operating in extreme scenarios

Challenge: Obtaining Real Data

Deployment of Sensors

- *Cost & sensor placement*

Recruitment of Participants

- *Reluctance to share data*
- *Time-consuming*
- *Limited in scale*

Preservation of Participant Privacy

- *Data regulations*
- *Leakage of sensitive data*

FERPA
Family Educational Rights and Privacy Act

HIPAA

GDPR

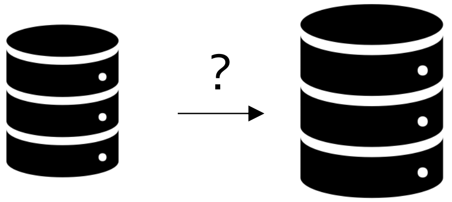
Generating Realistic Synthetic Data with Simulators

Challenge: Modeling smart spaces accurately

- Variability/dynamicity of activities
- Faithfulness to reality

Approach 1: Extend previously captured dataset[1]

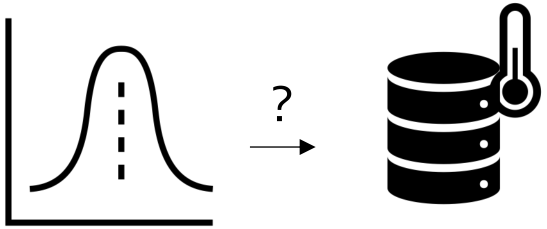
- Issue: violates causality, limited to initial space



Building X Building Y

Approach 2: Generate data randomly based on sensor models[2]

- Issue: random ≠ realistic



$20°C$ Temperature Data

Approach 3: Create dataset based on interactions of people and their activities[3]

- Issue: *Semantic Explainability* - Why people visit the spaces that they do?

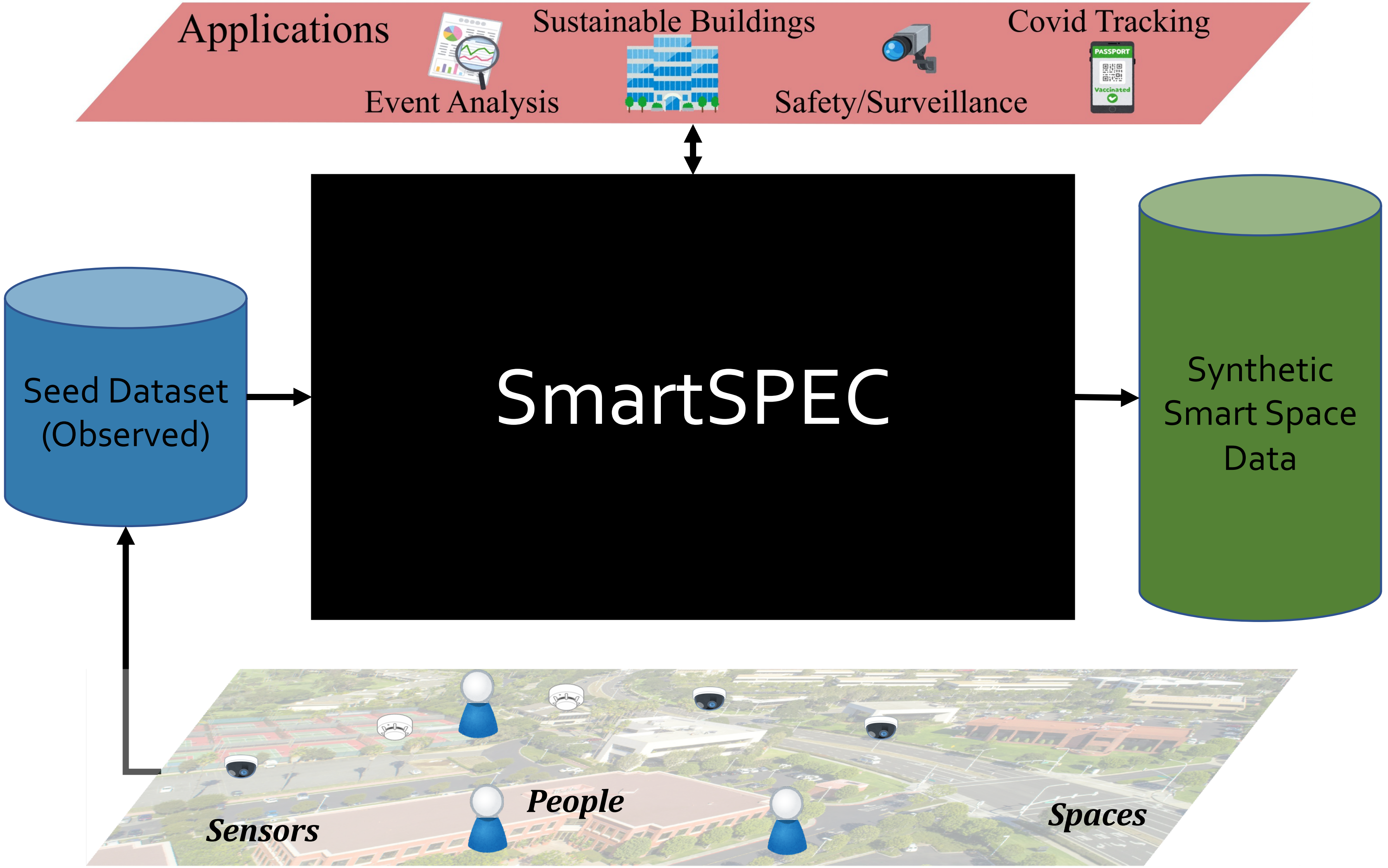
Brushing Toileting Walking

Activities of Daily Living

[1]**Replication, Modification, Sampling:** Tay et al., *UpSizeR* (Information Systems '13)
[2]**Random Data Generation:** *Mockaroo*, Hoag and Thompson, *PSDG* (ACM SIGMOD Record '07)
[3]**Activities of Daily Living**: Alshammari et al., *OpenSHS*, Sensors '17
Mobility Models and Trajectory Models: Rhee et al., IEEE/ACM TON '11; Alessandretti et al., Nature '20
Trajectory Models: Brinkoff, GeoInformatica '02; Pelekis et al., ACM Sigspatial '15
Generative Models: Gupta et al., CVPR '18; Rossi et al., Pattern Recognition '21

The SmartSPEC Approach

Exploit semantics to generate realistic synthetic smart space datasets



The Contributions of this Paper

SmartSPEC Platform

Seed Dataset (Observed)

Scenario Learning

Scenario Generation

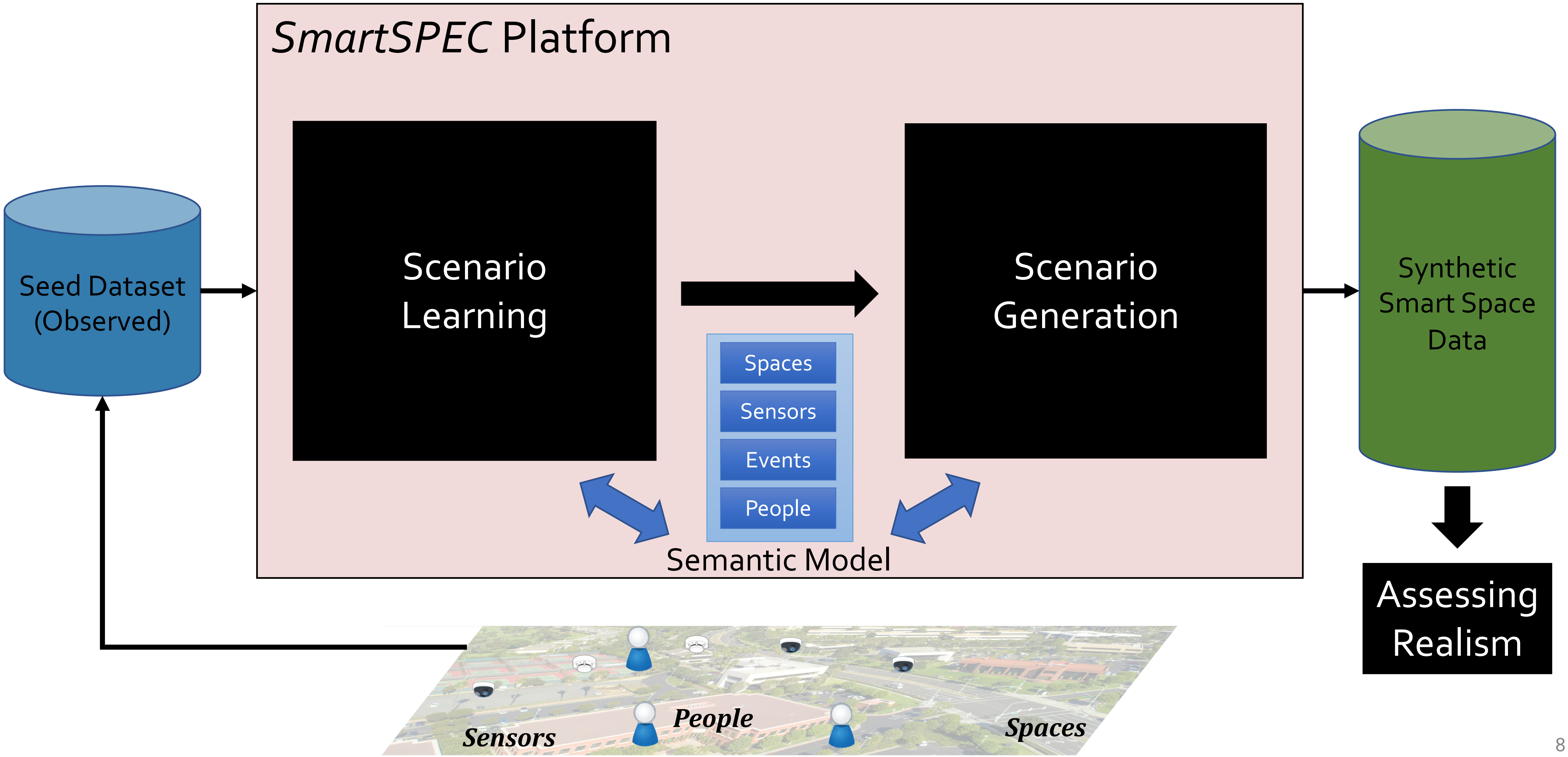
Semantic Model

Synthetic Smart Space Data

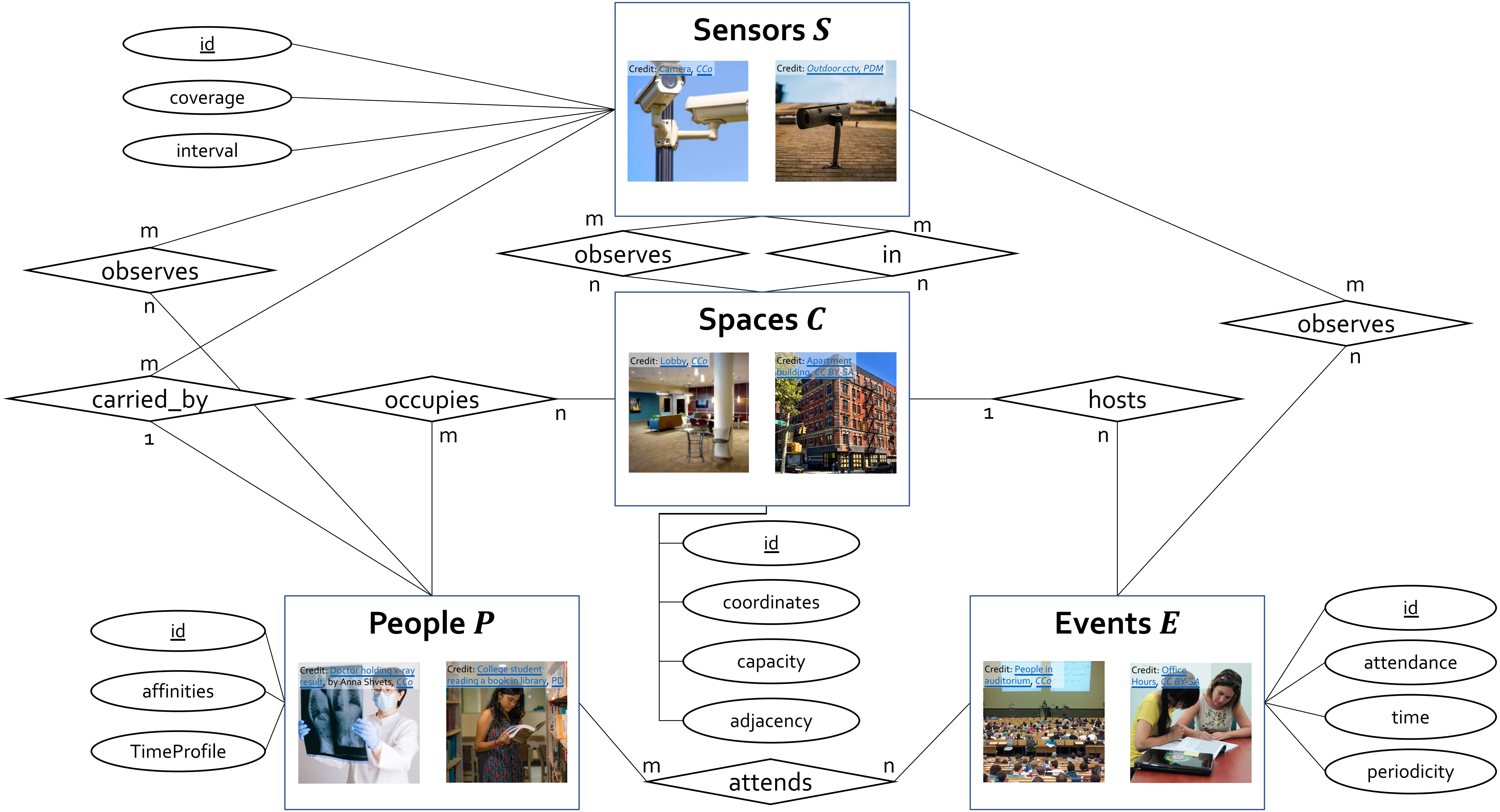
Assessing Realism

Scenario: a digital depiction of the activities and operations in the smart space

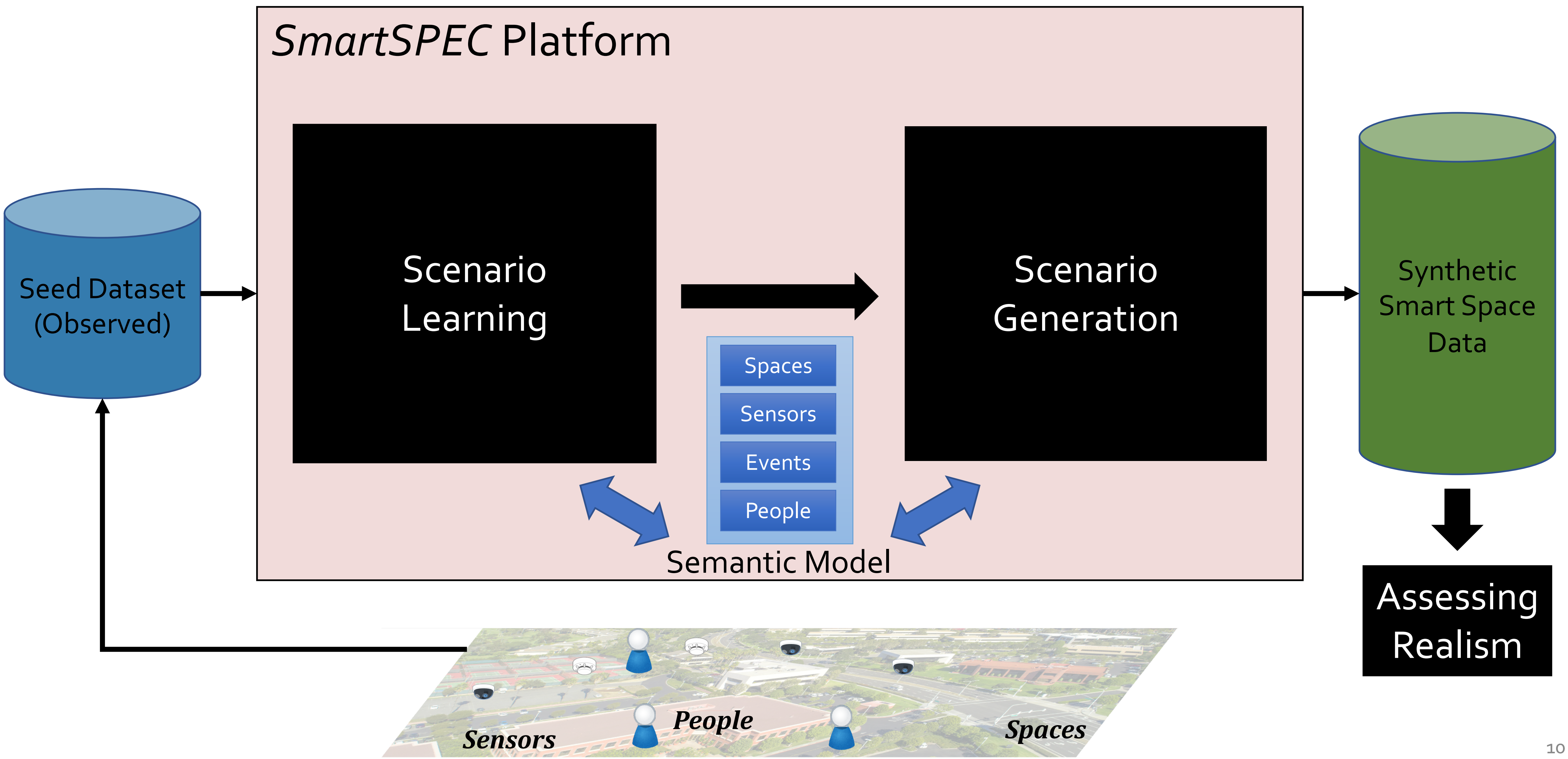
SmartSPEC : Semantic Model



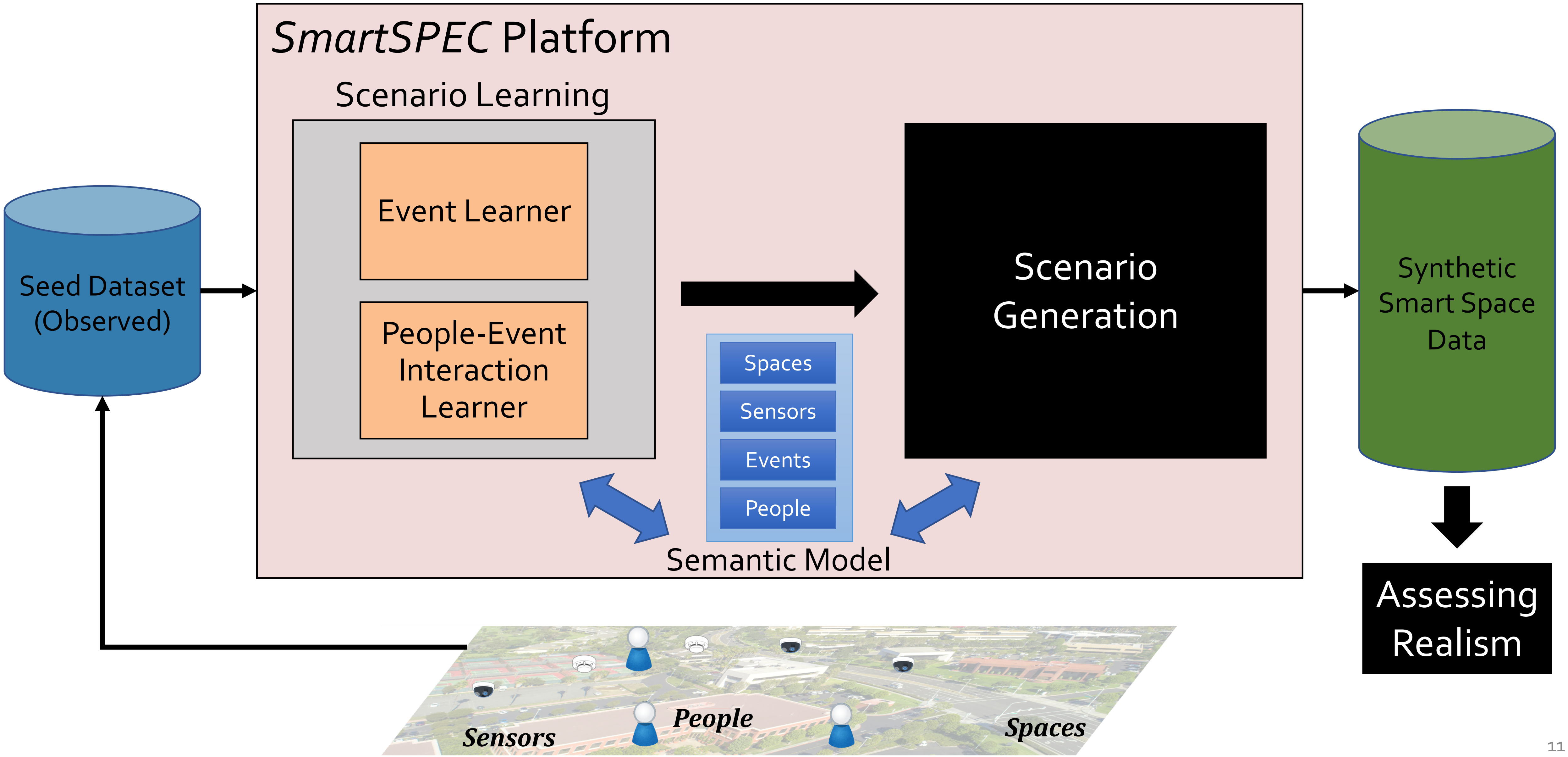
Smart Space: A Semantic Characterization



SmartSPEC : Scenario Learning



SmartSPEC : Scenario Learning



Learning Events through Occupancy

Dataset D

Person P	Space C	DateTime t
f28c94f	1412	2017-09-01 08:19:00
f20a461	6029	2017-09-01 08:19:00
238be6	3231	2017-09-01 08:19:07
238be6	3231	2017-09-01 08:19:26
…	…	…

For each space C

Person P	Space C	DateTime t
bb12b6	1100	2017-09-01 08:43:57
813a99	1100	2017-09-01 08:45:12
18bcad	1100	2017-09-01 08:45:38
81d9c1	1100	2017-09-01 08:46:20
81d9c1	1100	2017-09-01 08:46:23
500bba	1100	2017-09-01 08:47:23
f079e1	1100	2017-09-01 08:47:36
8700e1	1100	2017-09-01 08:47:49
84ea3f	1100	2017-09-01 08:48:21
500bba	1100	2017-09-01 08:49:38
…	…	…

Occupancy λ_D^{c,t_s,t_e}

1

3

4

…

Occupancy λ_D^{C,t_s,t_e}

- Number of unique people from dataset D that are in space C during time period (t_s, t_e).

Algorithm 1: Extracting Events, Learning MetaEvents.

Input: Dataset D, Spaces $\mathcal{C}$, Date $start$, Date end, int b
Output: Events $\mathcal{E}$, MetaEvents $\mathcal{ME}$

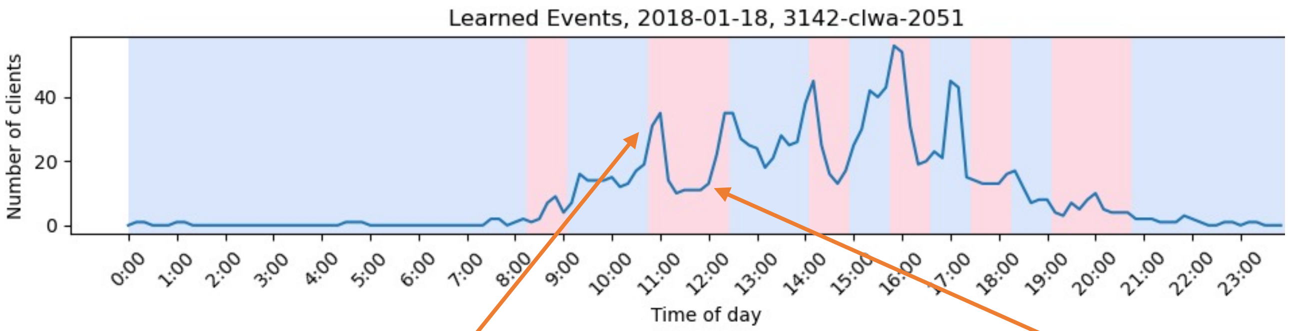
$\mathcal{E} \leftarrow \emptyset$
for $d \leftarrow start \ldots end$ **do**
  for $c \leftarrow \mathcal{C}$ **do**
    $data \leftarrow D.query(space = c, day = d)$
    $ts \leftarrow computeOccupancy(data, minutes = b)$
    $bkpts \leftarrow changePointDetection(ts)$
    $\mathcal{E} \leftarrow \mathcal{E} \cup createEvents(c, bkpts)$
$distMat \leftarrow computeDistanceMatrix(\mathcal{E})$
$clusters \leftarrow doAgglomerativeClustering(distMat)$
$\mathcal{ME} \leftarrow makeMetaEvents(clusters)$
return $\mathcal{E}$, $\mathcal{ME}$

Intuition:

Create time-series of occupancy in space C on date d

Use *Change Point Detection* to learn when one event ends, and another starts

Intuition:
Change Point Detection



Breakpoints occur when there are large changes in occupancy

Occupancy stays roughly consistent during an event

Presence → Occupancy → Events

Learning Events

Algorithm 1: Extracting Events, Learning MetaEvents.

```
Input: Dataset D, Spaces 𝒞, Date start, Date end, int b
Output: Events ℰ, MetaEvents ℳℰ
ℰ ← ∅
for d ← start ... end do
    for c ← 𝒞 do
        data ← D.query(space = c, day = d)
        ts ← computeOccupancy(data, minutes = b)
        bkpts ← changePointDetection(ts)
        ℰ ← ℰ ∪ createEvents(c, bkpts)
distMat ← computeDistanceMatrix(ℰ)
clusters ← doAgglomerativeClustering(distMat)
ℳℰ ← makeMetaEvents(clusters)
return ℰ, ℳℰ
```

Intuition:

Create time-series of occupancy in space C on date d

Use *Change Point Detection* to learn when one event ends, and another starts

Use *Agglomerative Clustering* to learn types of events

Intuition: *Agglomerative Clustering*

- Each event starts in its own cluster, and is merged with other "nearby" clusters
- Terminates once distance between clusters $\geq$ threshold ϵ
- Cluster distance based on set of attendees and time of event

Jaccard Index

- Given two sets A and B, define similarity ratio $r = \frac{card(A \cap B)}{card(A \cup B)}$.
- *Interpretation*: $r = 1$ only if $A = B$.

Presence → Occupancy → Events

Learning People-Event Interactions

Learned Events:

- Event e_1: attendees = $\{p_1, p_2, p_3\}$
- Event e_2: attendees = $\{p_2, p_3\}$
- Event e_3: attendees = $\{p_1\}$
- Event e_4: attendees = $\{p_3\}$
- Event e_5: attendees = $\{p_1, p_2\}$

Characterize people based on attended events

Person p_1

attended: $\{e_1, e_3, e_5\}$

Person p_2

attended: $\{e_1, e_5\}$

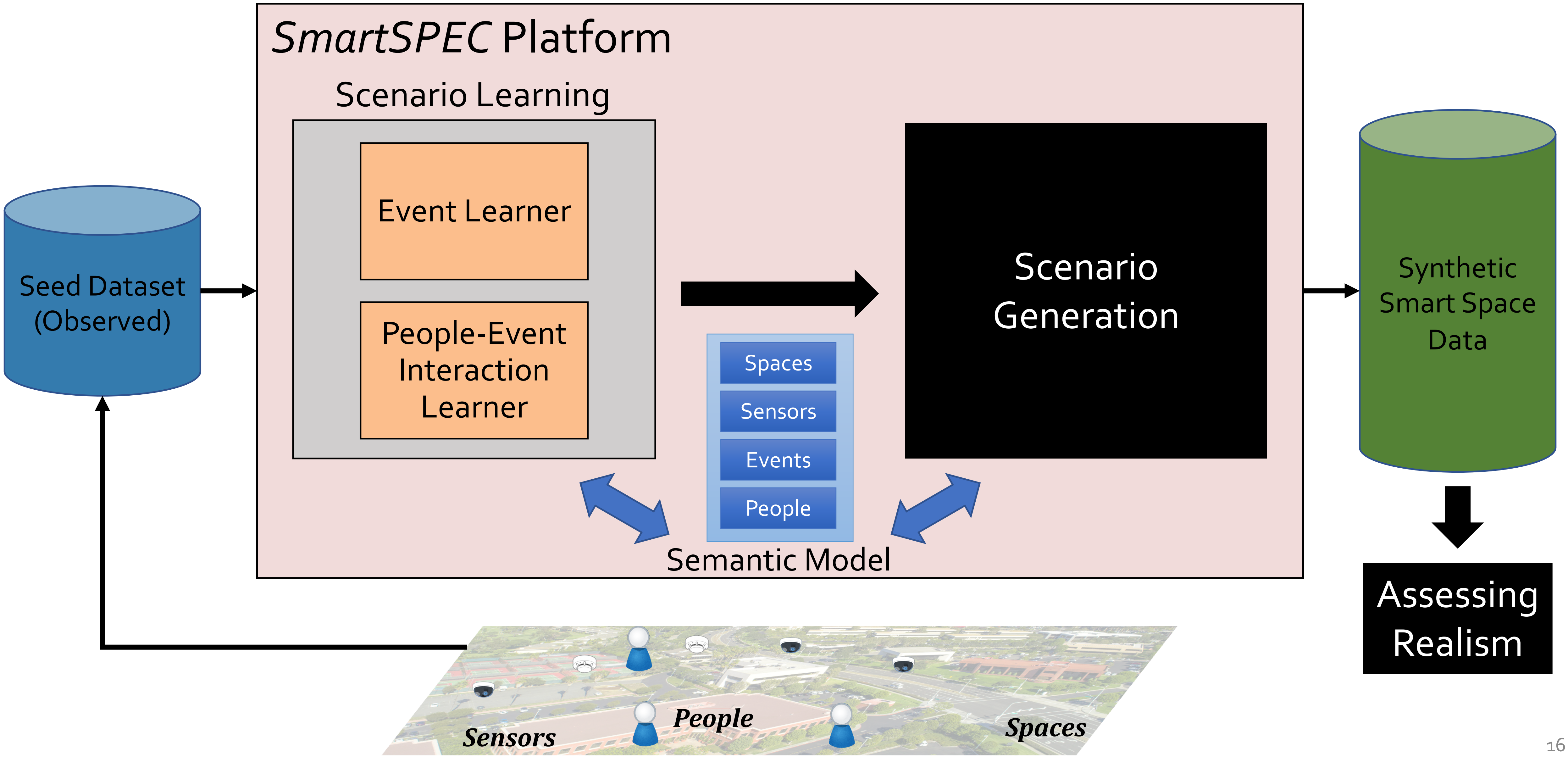
Person p_3

attended: $\{e_1, e_2, e_4\}$

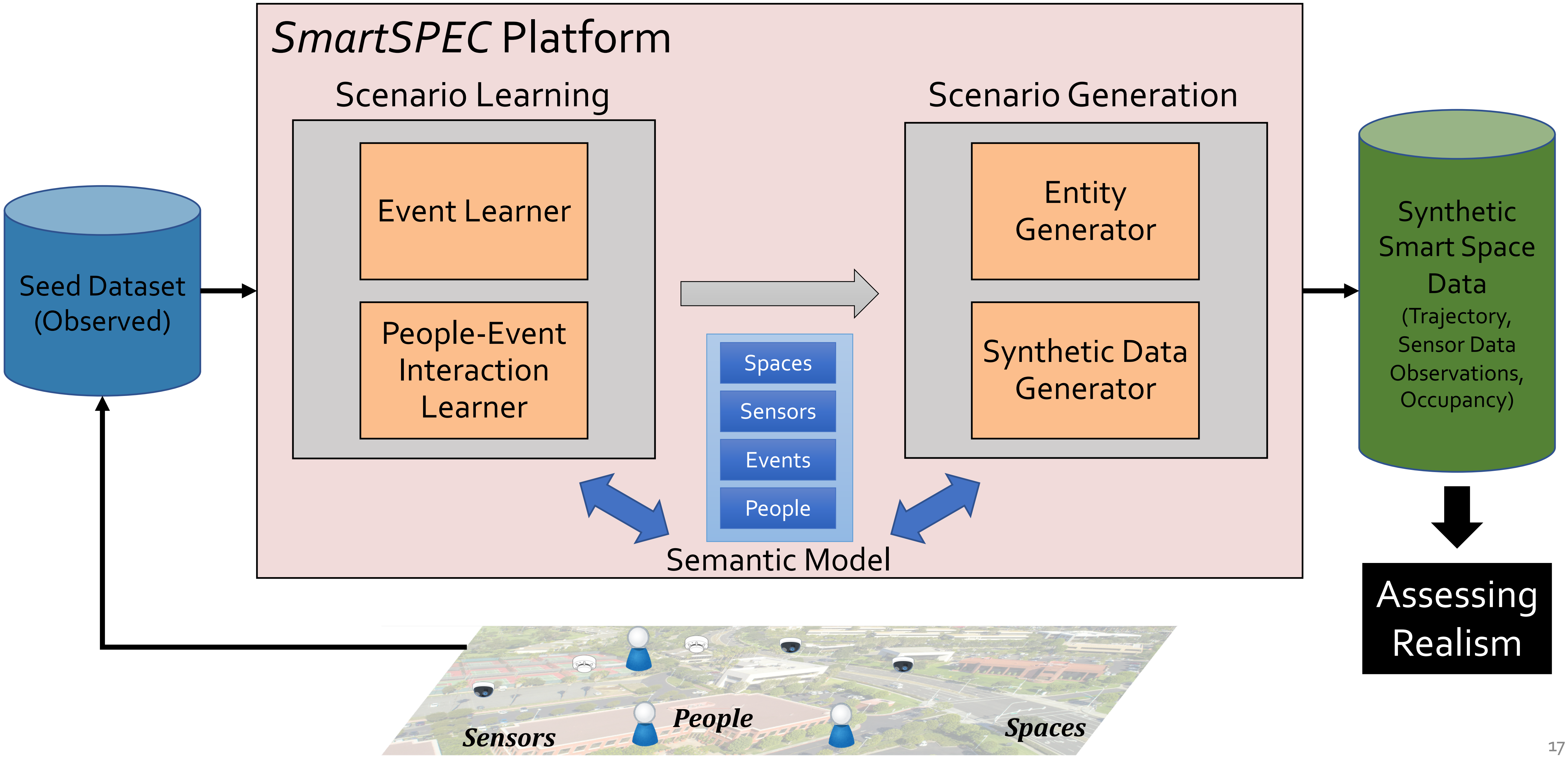
…

Apply *Agglomerative Clustering* to group people by similarity of attended events (until a threshold ϵ)

SmartSPEC : Scenario Generation

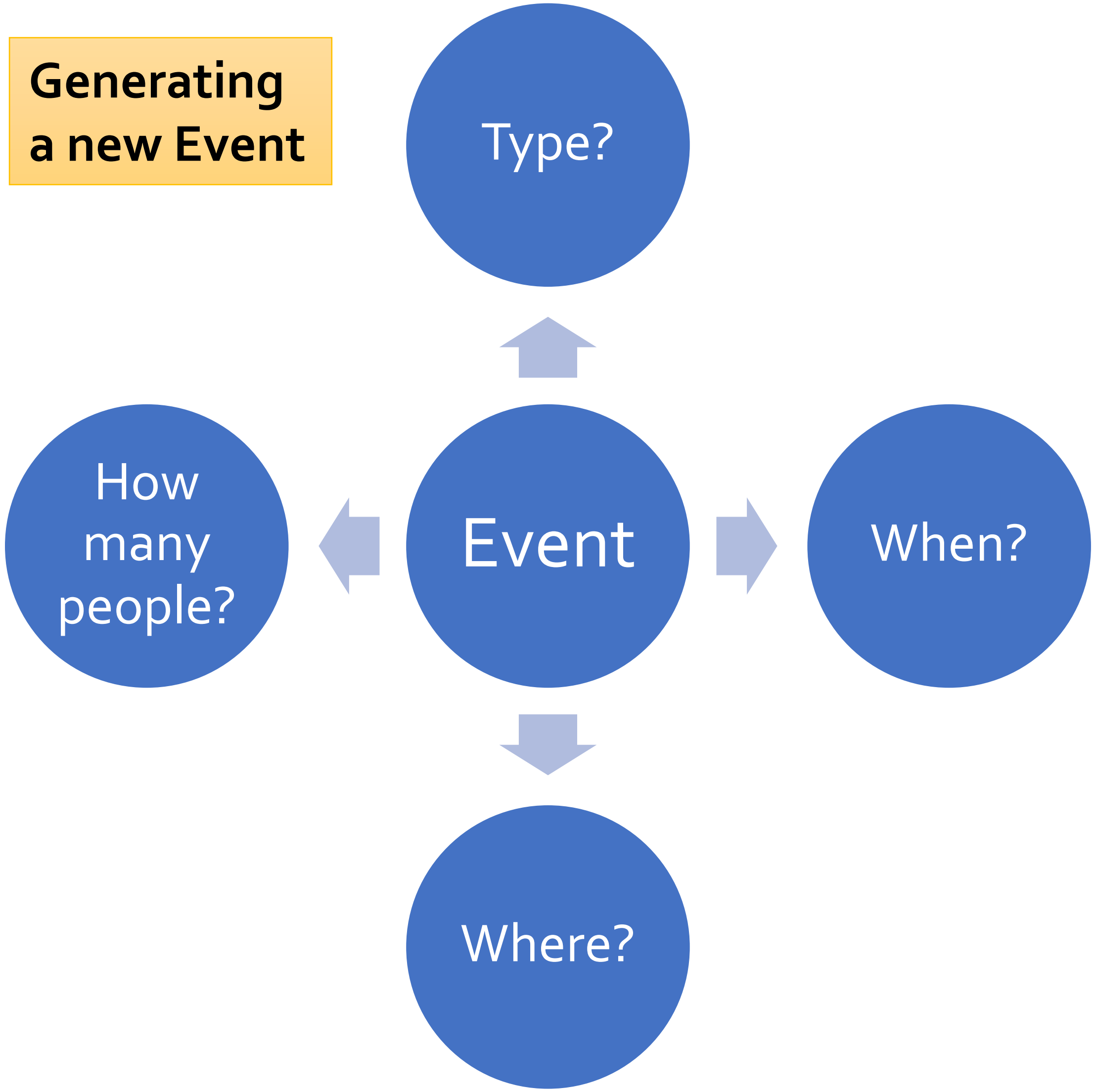


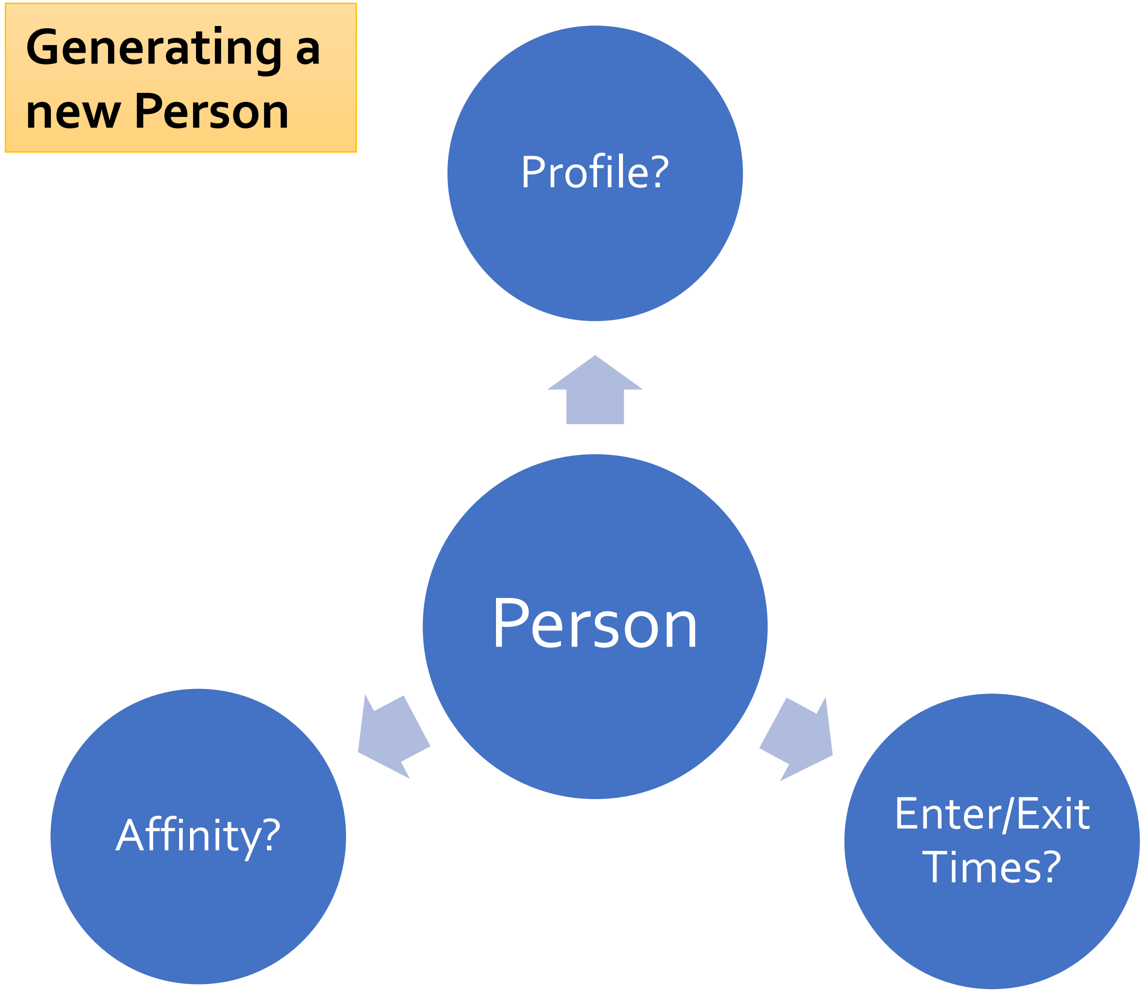
SmartSPEC : Scenario Generation



Entity Generator: Generating Events and People

Given types of events and profiles of people, how can we create a new set of events and people for our synthetic dataset?





Synthetic Data Generator: Generating Synthetic Data

Algorithm 3: Synthetic data generation.

Input: Date d_s, Date d_e, People $\mathcal{P}$, Events $\mathcal{E}$, Spaces $\mathcal{C}$
Output: LogFile log

$log \leftarrow \emptyset$
for $P \leftarrow \mathcal{P}$ **do**
  for $d, t_s, t_e \leftarrow P.queryActiveDateTime(d_s \dots d_e)$ **do**
    $t \leftarrow t_s$
    while $t \leq t_e$ **do**
      $E \leftarrow P.findPreviousEvent(d, t)$
      if ! E *is* **null then**
        $path \leftarrow getPath(P.space, E.space)$
      else
        $attd \leftarrow \emptyset$
        for $E \leftarrow \mathcal{E}$ **do**
          if $!E.hasSpaceCapacity(t)$ **or** $!E.hasPeopleCapacity(P)$ **or** $E.conflictsWith(P.prevEvents)$ **then**
            continue
          $P_e \leftarrow getPath(P.space, E.space)$
          $arrival \leftarrow t + P_e.estTravelTime()$
          if $|arrival - E.startTime| \geq \epsilon$ **then**
            continue
          $attd \leftarrow attd \cup \{(\mathcal{E}, P_e)\}$
        $E, path \leftarrow select(attd, P.eventAffinity)$
      for $c \leftarrow path$ **do**
        Block until $C_e.cap(d, t) \leq C_e.maxCap$
        Move P to c, updating t
        $log.record(P, c, t)$
      $log.record(P, E.space, E.t_e)$
      $P.recordAttendance(E)$
      $t \leftarrow E.t_e$
return log

Intuition:

Get date/time that person is in the smart space

Choose an event to attend, preferably a previously attended periodic event

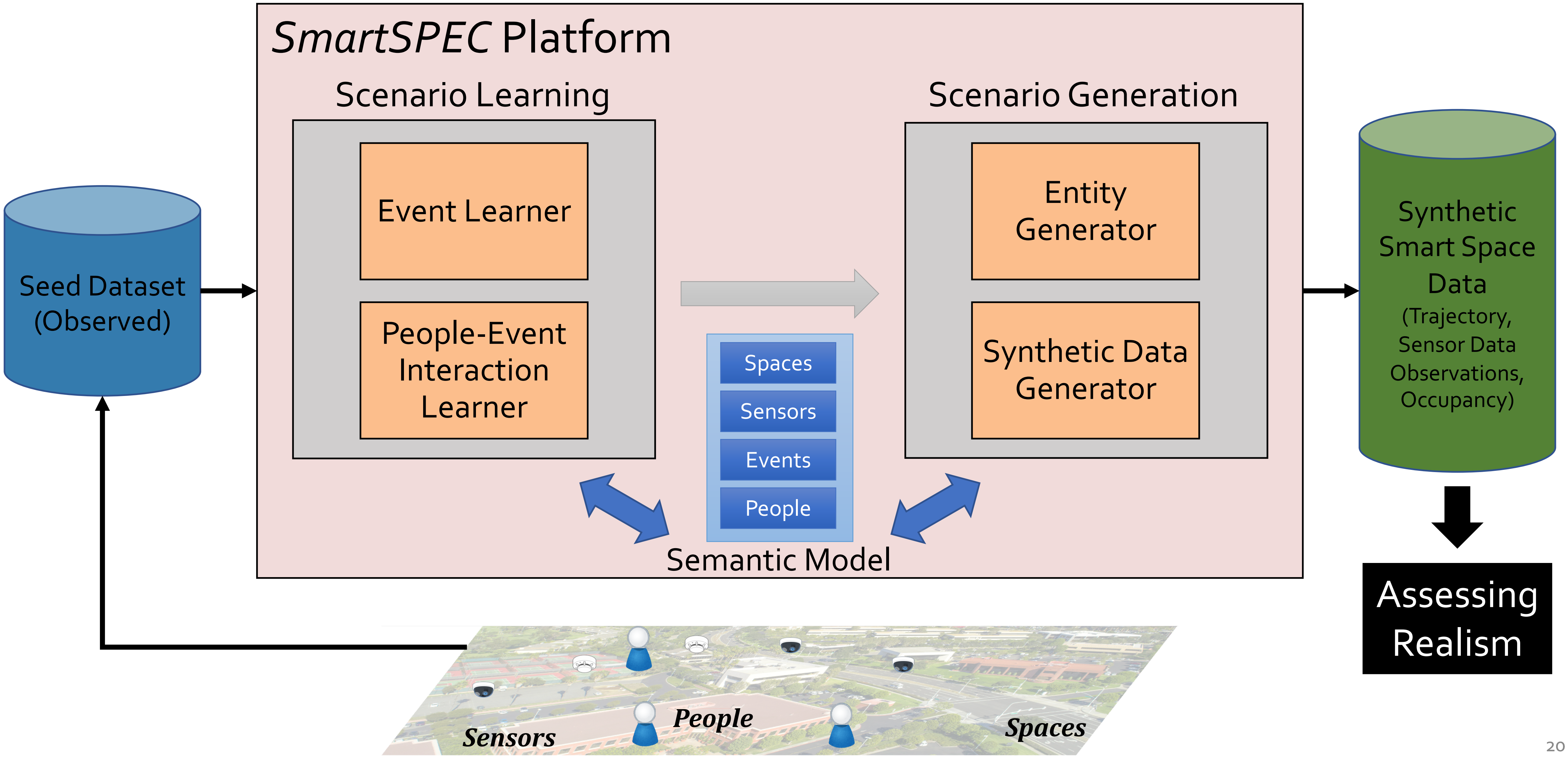
Semantic Constraints on spaces, people, events

Estimate travel time; estimated arrival must be within a threshold ϵ

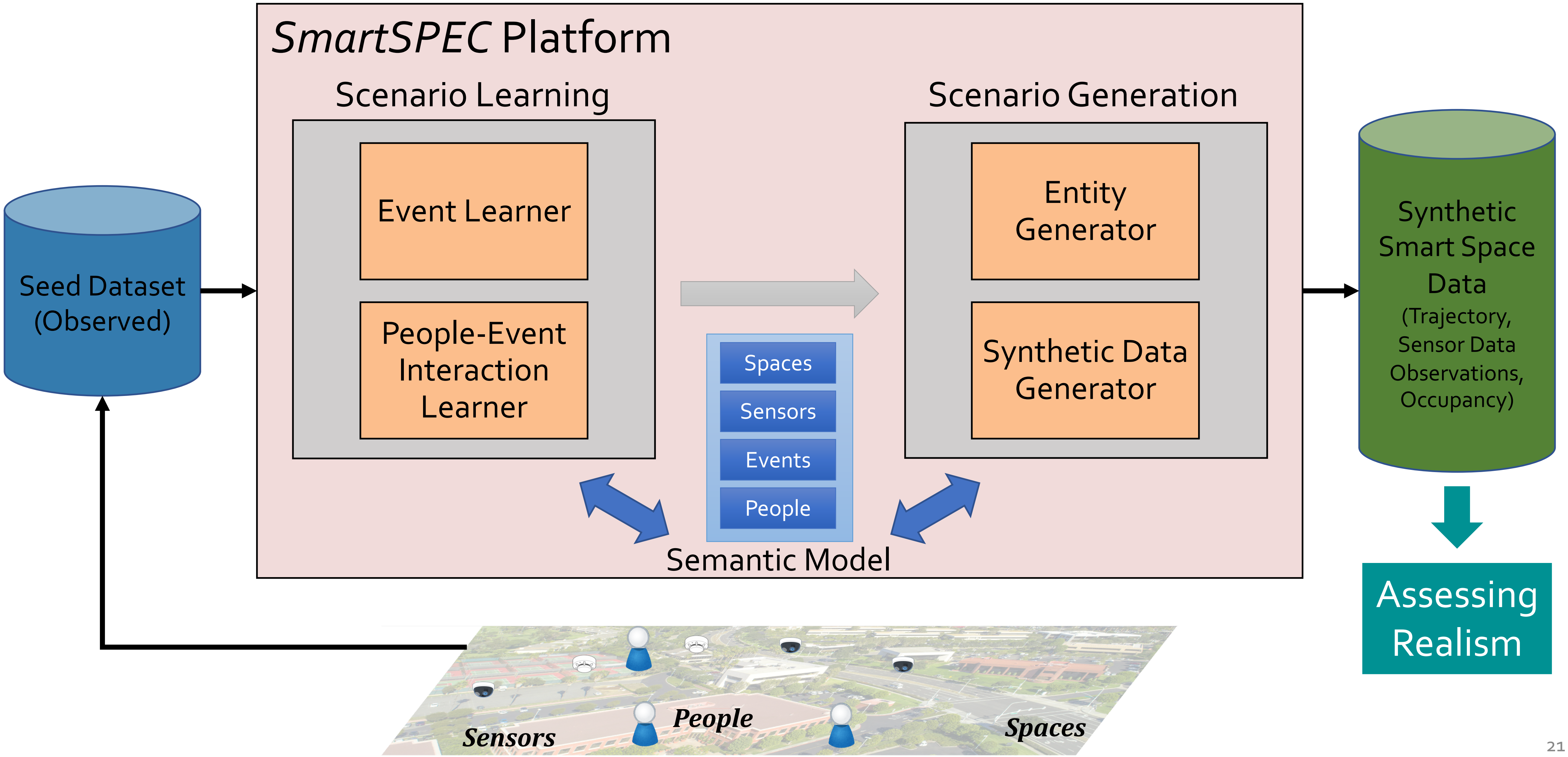
Move to an event space

Record data in log file

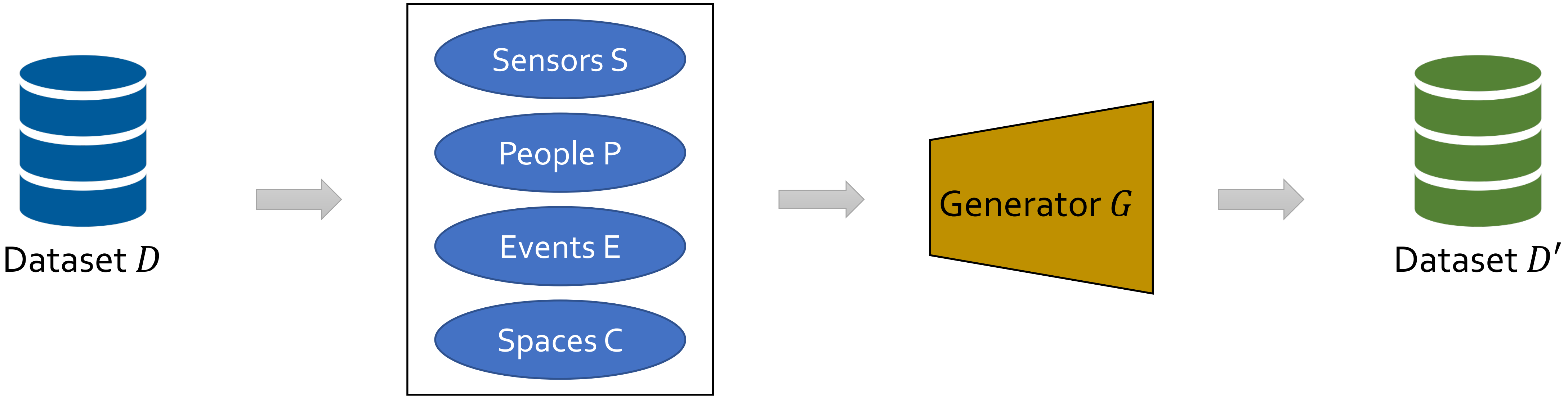
SmartSPEC : Assessing Realism



SmartSPEC : Assessing Realism



Assessing Realism of Smart Space Datasets



Person P	Space C	DateTime t
f28c94f	1412	2017-09-01 08:19:00
f20a461	6029	2017-09-01 08:19:00
238be6	3231	2017-09-01 08:19:07
238be6	3231	2017-09-01 08:19:26
…	…	…

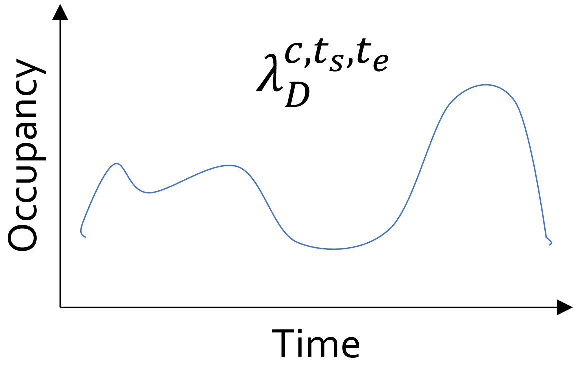
How to quantify the realism of D, D'?

- *Occupancy*: a space's perspective of the dataset
- *Trajectory*: a person's perspective of the dataset

Similarity of Space's Occupancy

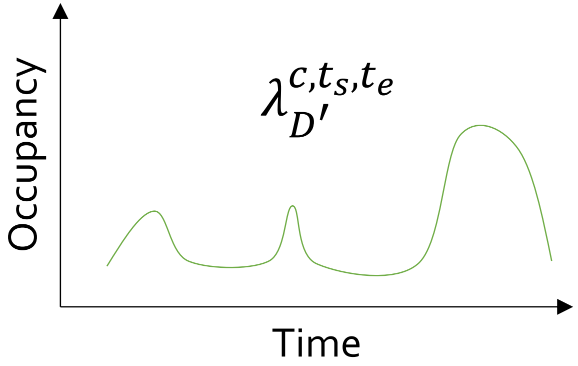
Dataset D

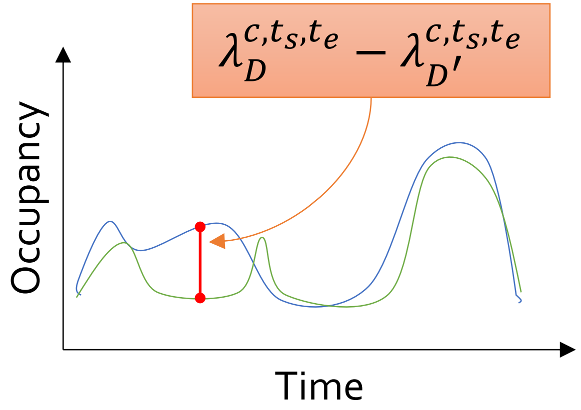
Extract Occupancy



Dataset D'

Extract Occupancy



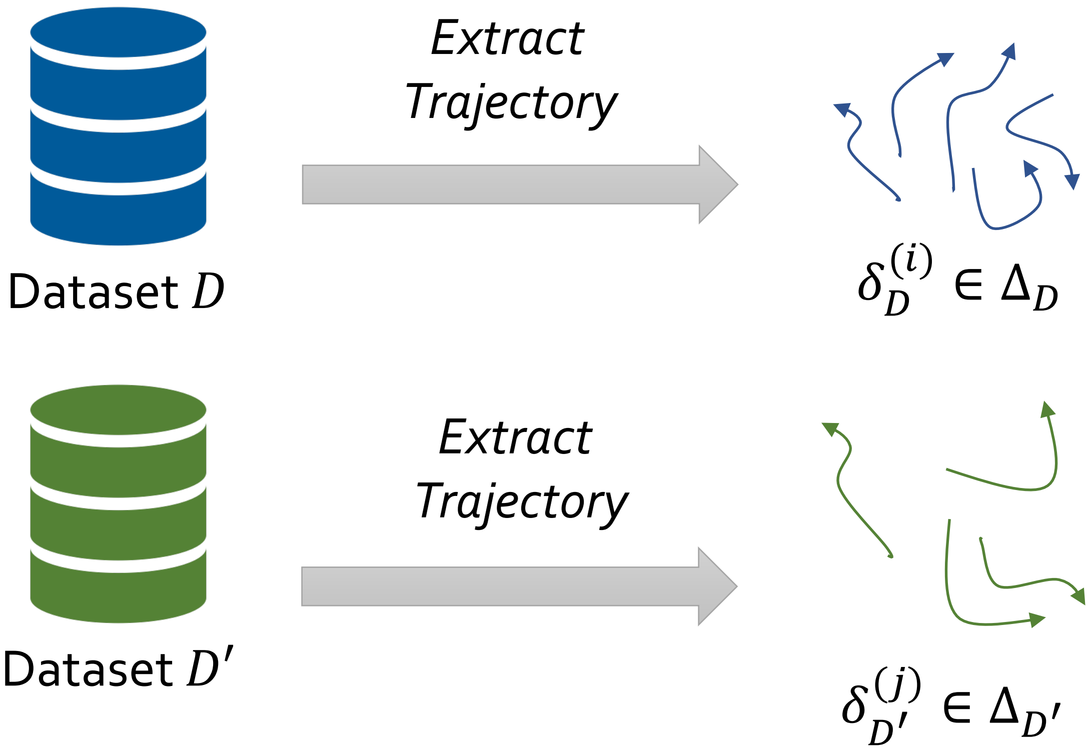


Occupancy Distance

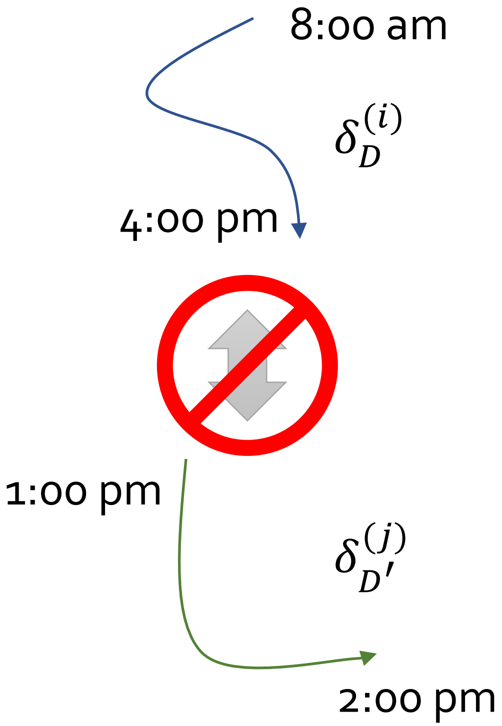
$$\frac{1}{|C||t_s - t_e|} \sum_{\substack{c \in C \\ t_s, t_e \in T}} \left| \lambda_D^{c,t_s,t_e} - \lambda_{D'}^{c,t_s,t_e} \right|^2$$

- *Occupancy* of space C: number of unique people in space C during time period (t_s, t_e).
- *Occupancy Distance* is the mean squared error in occupancy over time.

Similarity of People's Trajectory



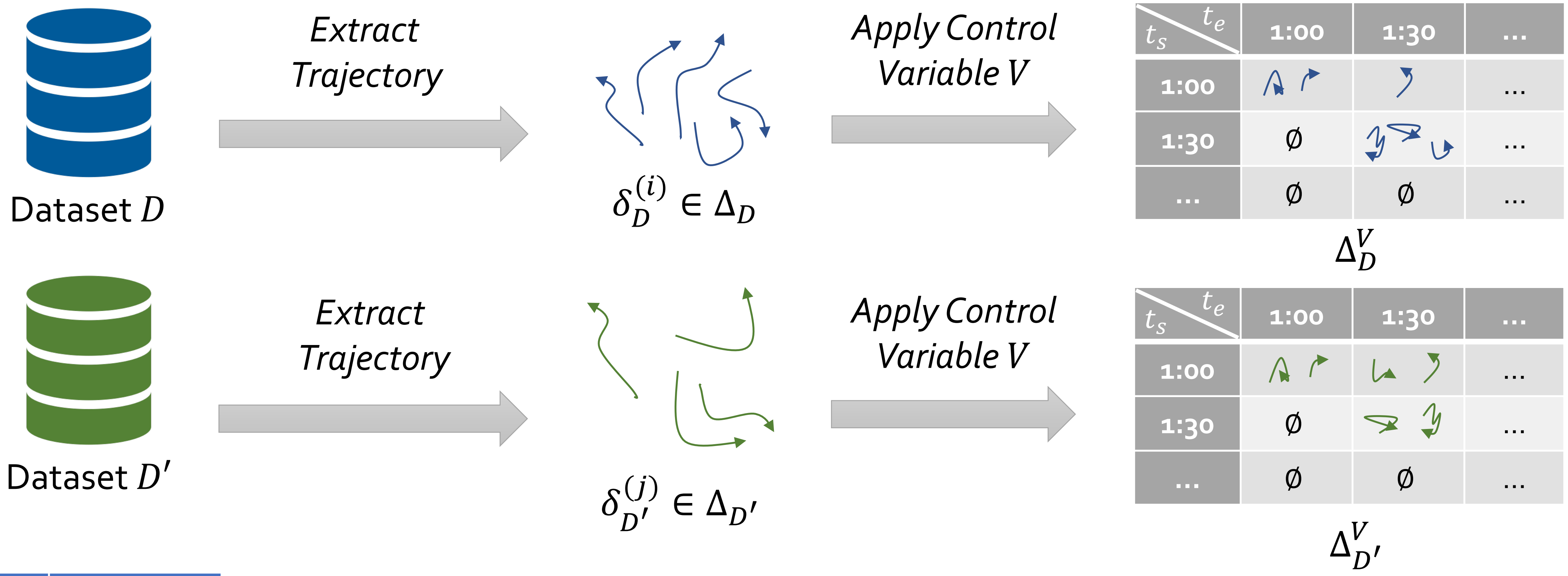
Consider the following:



Person P	Space C	DateTime t
238be6	3231	2017-09-01 08:19:07
238be6	3231	2017-09-01 08:19:26
238be6	3254	2017-09-01 08:20:50
238be6	3256	2017-09-01 08:21:13
…	…	…

- *Trajectory* of person P: sequence of spaces C visited by P over datetime t
  - *Should we naïvely compare all trajectories against each other?*

Similarity of People's Trajectory



Person P	Space C	DateTime t
238be6	3231	2017-09-01 08:19:07
238be6	3231	2017-09-01 08:19:26
238be6	3254	2017-09-01 08:20:50
238be6	3256	2017-09-01 08:21:13
...	...	...

- **Control Variables** are applied to *partition* trajectories into comparable bins. e.g., $V = (t_s, t_e)$ = (1:00, 1:30) contains trajectories with $t_s \approx$ 1:00, $t_e \approx$ 1:30.

Similarity of People's Trajectories

Distance Function Φ

t_s \ t_e	1:00	1:30	…
1:00			…
1:30	∅		…
…	∅	∅	…

Δ_D^V

$t_s, t_e = (1{:}00, 1{:}00)$

$\Phi(\quad,\quad)$

$t_s, t_e = (1{:}00, 1{:}30)$

$\Phi(\quad,\quad)$

$t_s, t_e = (1{:}30, 1{:}30)$

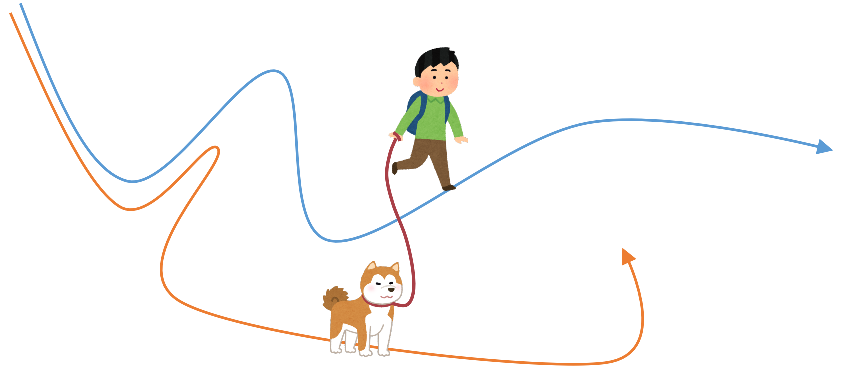
$\Phi(\quad,\quad)$

t_s \ t_e	1:00	1:30	…
1:00			…
1:30	∅		…
…	∅	∅	…

$\Delta_{D'}^V$

Distance Function Φ

- Let $\phi(\delta_D^{(i)}, \delta_{D'}^{(j)})$ be a function that computes the distance between two trajectories
- e.g., Fréchet Distance Metric



How do we compare multiple trajectories against one another?

Similarity of People's Trajectories

Distance Function Φ

$t_s \backslash t_e$	1:00	1:30	…
1:00			…
1:30	$\emptyset$		…
…	$\emptyset$	$\emptyset$	…

Δ_D^V

$t_s \backslash t_e$	1:00	1:30	…
1:00			…
1:30	$\emptyset$		…
…	$\emptyset$	$\emptyset$	…

$\Delta_{D'}^V$

$t_s, t_e = (1{:}00, 1{:}00)$

$\Phi(\quad,\quad)$ $\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$

$t_s, t_e = (1{:}00, 1{:}30)$

$\Phi(\quad,\quad)$ $\begin{matrix} 1 & 1 \\ 0 & 0 \end{matrix}$

$t_s, t_e = (1{:}30, 1{:}30)$

$\Phi(\quad,\quad)$ $\begin{matrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{matrix}$

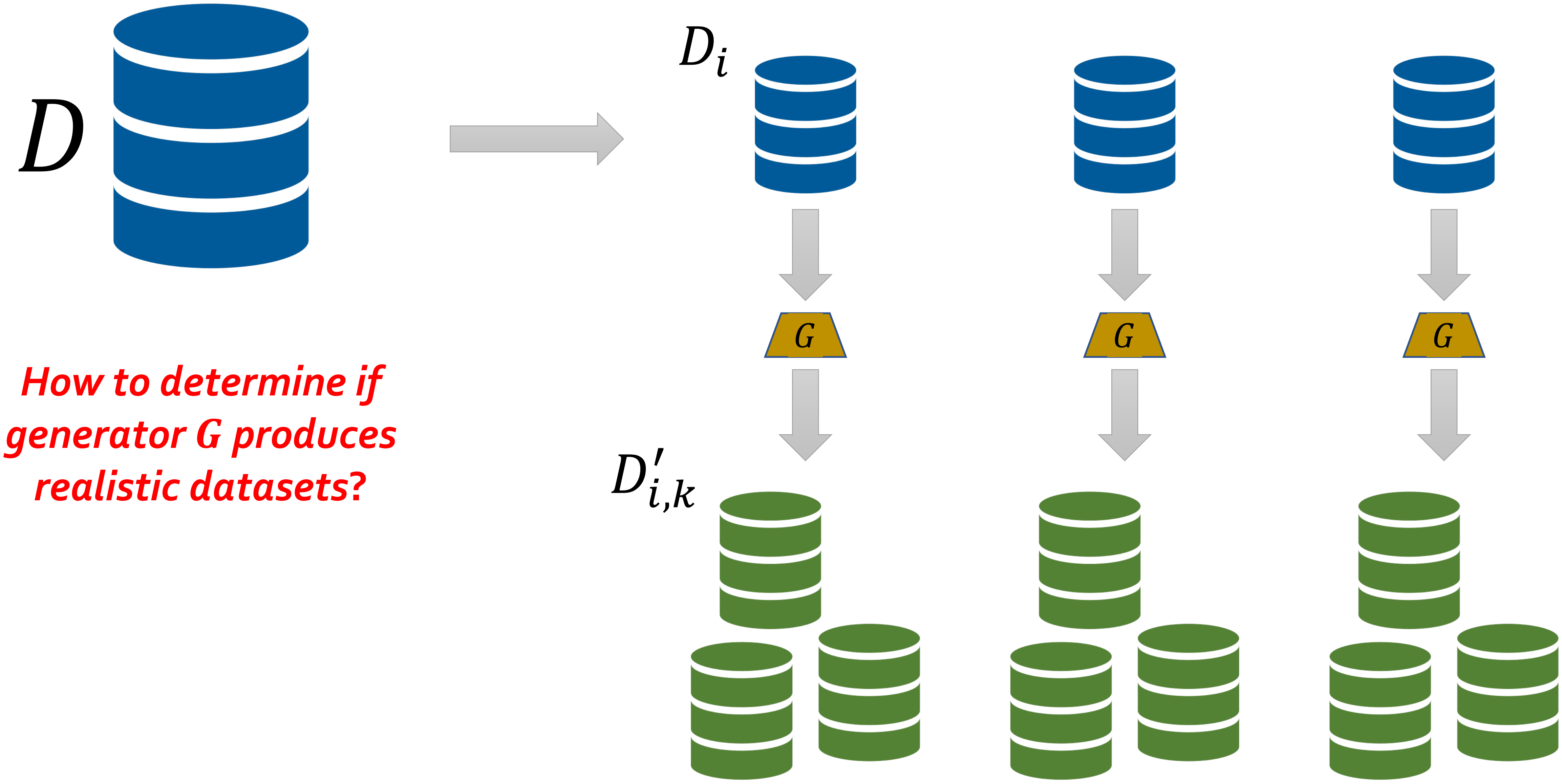
Trajectory Distance

$$\frac{1}{|V|} \sum_{\substack{v \in V \\ (\delta^{(i)}, \delta^{(j)}) \in M}} \Phi(\delta^{(i)}, \delta^{(j)}) + \alpha\left(|\Delta_D^v| - |\Delta_{D'}^v|\right)$$

Penalty Term for difference in trajectory set sizes

- **Match** trajectories between corresponding bins
- Matching matrix M does not need to be injective

Interpreting Dataset Similarity

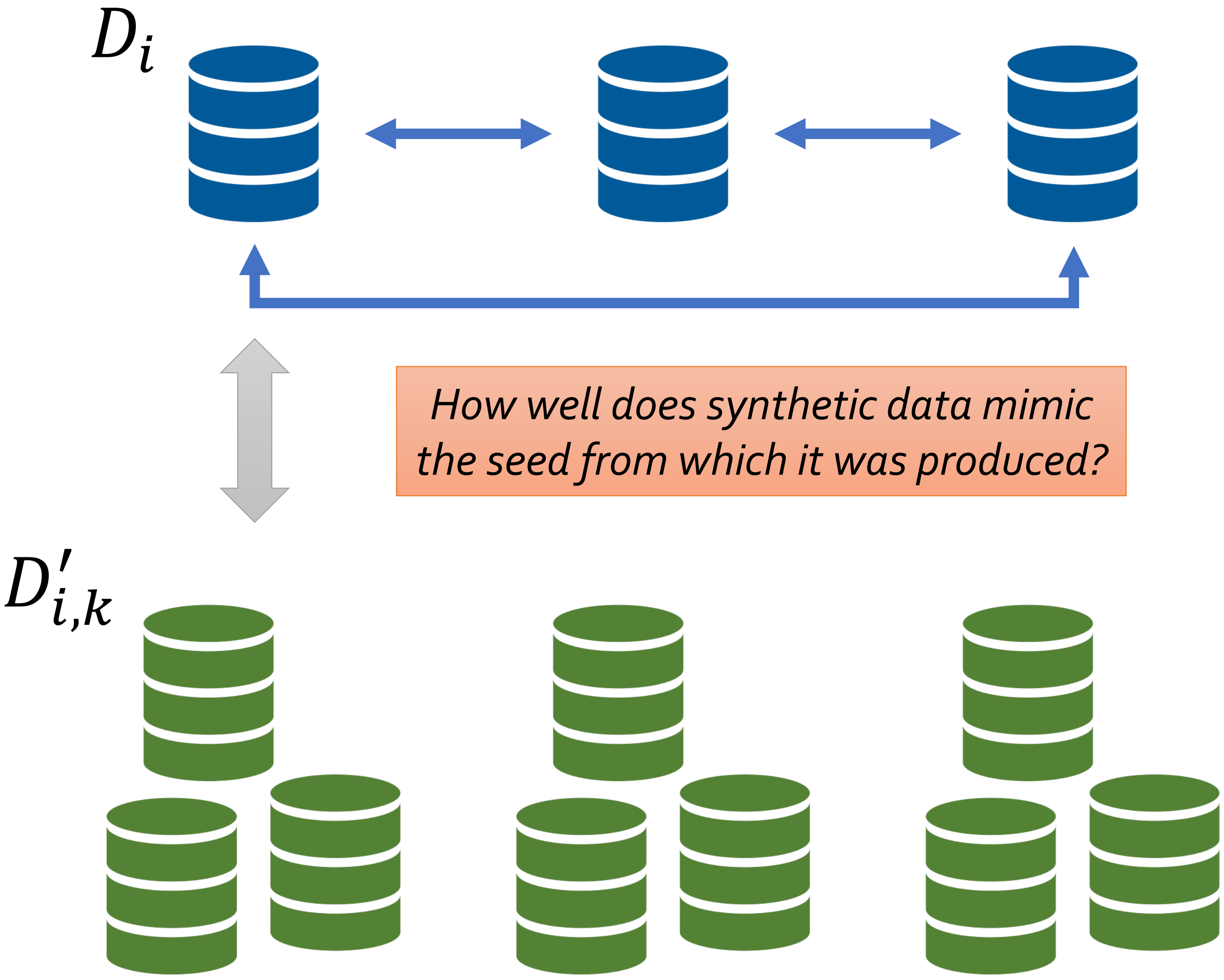


How to determine if generator G produces realistic datasets?

Interpreting Dataset Similarity

Compare distances between pairs of real datasets

*How do **real** datasets vary against other **real** datasets?*



Compare distances between pairs of real and simulated datasets

*How do **real** datasets differ from **synthetic** datasets?*

Interpreting Dataset Similarity

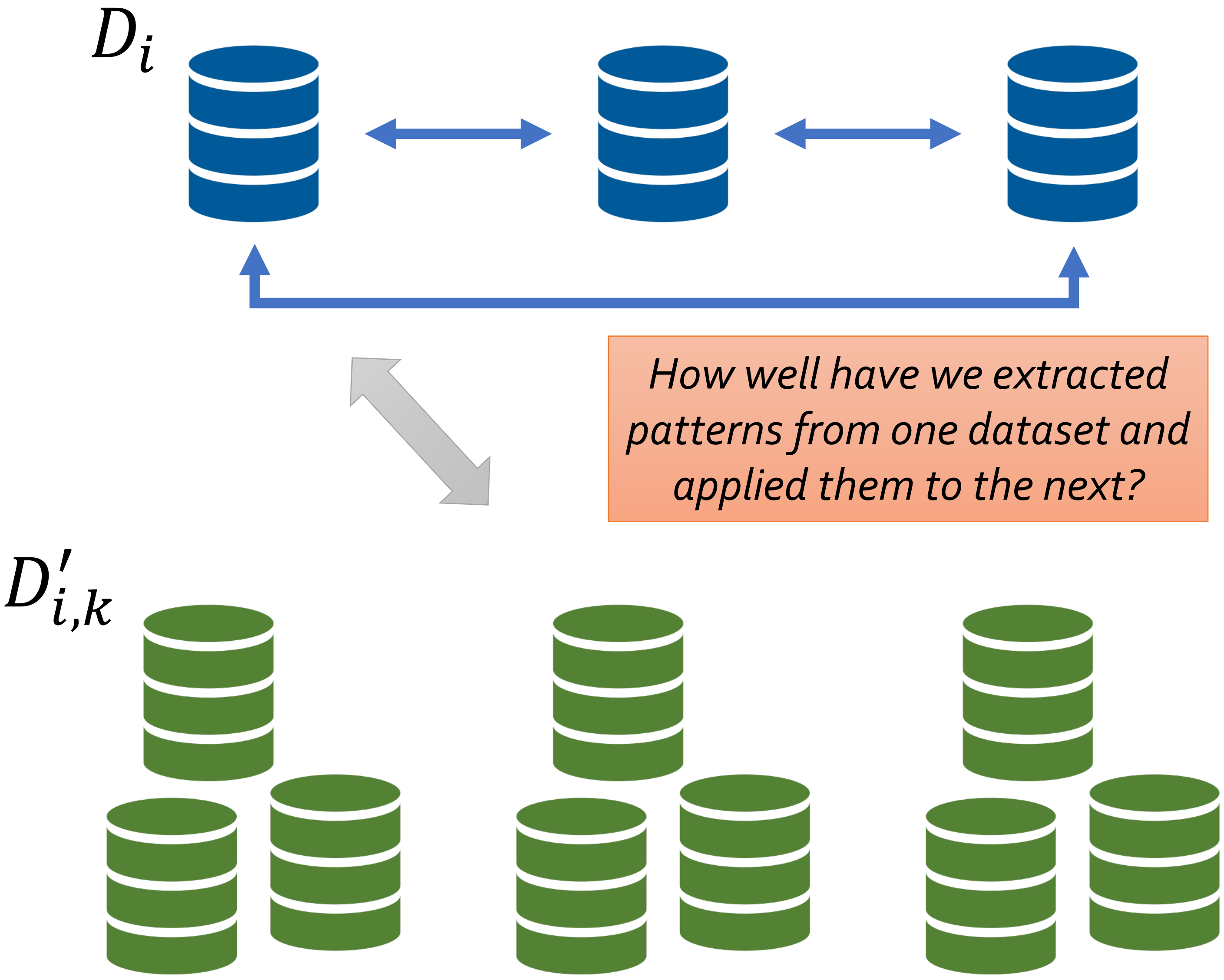
Compare distances between pairs of real datasets

*How do **real** datasets vary against other **real** datasets?*

Simulated ≈ Real?

Compare distances between pairs of real and simulated datasets

*How do **real** datasets differ from **synthetic** datasets?*



How well have we extracted patterns from one dataset and applied them to the next?

Experiment: 2 Distinct Scenarios

Scenario 1: Campus

- 6 floor campus building: 125+ faculty offices, 10 classrooms, 4 lecture halls
- 64 WiFi Access Points (WiFi APs)
- 5 weeks of WiFi connectivity events, ~300K connections/week, partitioned into 5 periods of 1 week each

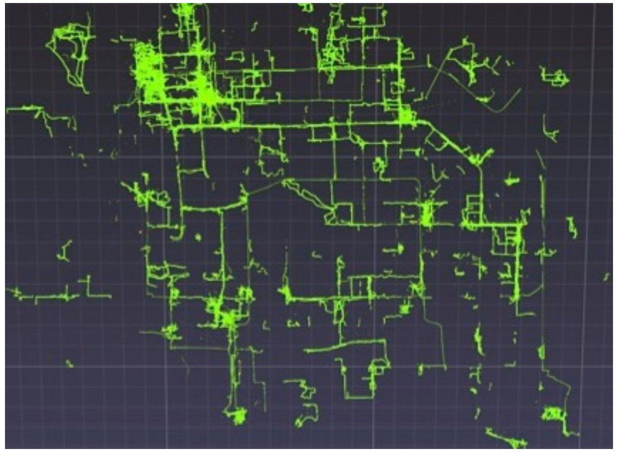
Scenario 2: City – GeoLife GPS Trajectories[1]

- GPS trajectories in city of Beijing, China
- 1150 points of interest to cluster GPS data
- 63 people over 28 months, ~36K GPS data/month, partitioned into 1-month periods

Bren Hall, UC Irvine

1st Floor Blueprint

Beijing, China



GeoLife GPS Trajectories

Learned types of events / profiles of people from both scenarios

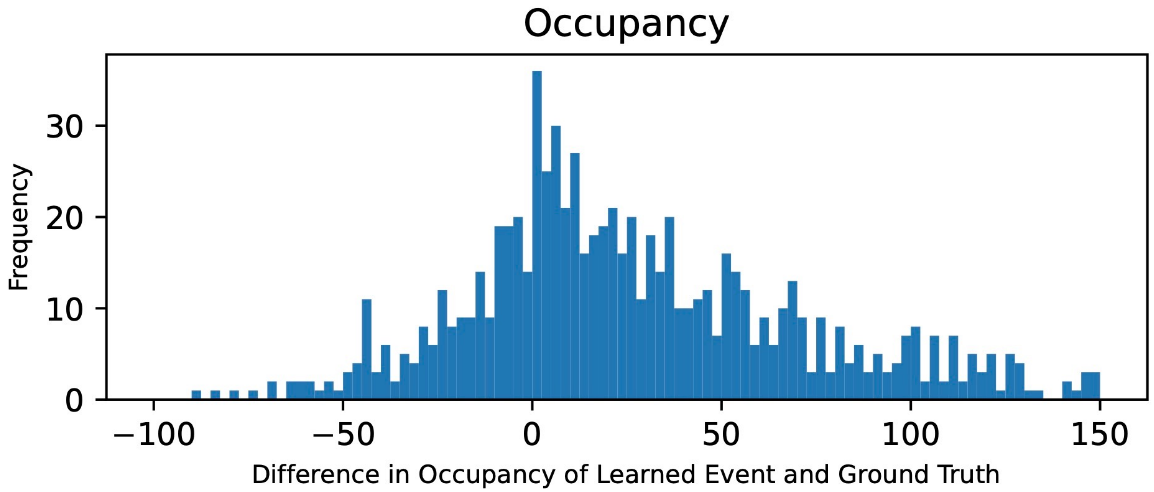
[1] Zheng et al., "Geolife: A collaborative social networking service among user, location and trajectory." IEEE Data Eng. Bull., vol. 33, no. 2.

Learned Events in Campus Scenario

Events

- 510 "ground truth" events
- Best-effort mapping of events to WiFi APs

- Average paired difference between:
  - Event Start Time: $15 \pm 18\ mins$
  - Event End Time: $21 \pm 27\ mins$



Baselines and Metrics

Mobility Model Baselines

- *Random Waypoint (RAND)*: Next visited space is random
- *Brownian Motion (BROW)*: Next visited space is adjacent
- *Lévy Flight (LÉVY)*: Next visited space is chosen by following a power law distribution on distance
- *Exponential Preferential Return (EPR)*: Same as Lévy Flight but selects previously visited spaces with higher probability

Comparison Metrics

- *Trajectory Distance*: Average paired Fréchet distance controlled over start/end times
    - Start/End Times on 30-minute blocks
- *Occupancy Distance*: Average difference in occupancy
    - Over 5-minute intervals
- Averaged results from 3 simulations, comparing against next week (campus scenario) or month (city scenario)

Evaluating Realism in Campus Scenario

Campus Scenario

	Week 1	Week 2	Week 3	Week 4
Real	185.65	188.67	191.31	194.60
SmartSPEC	263.92	252.09	272.43	240.99
RAND	789.8	754.07	740.23	606.74
BROW	533.27	479.68	501.39	407.32
LÉVY	760.3	713.53	713.18	583.97
EPR	693.38	554.26	635.81	459.4

Trajectory Similarity (m)

	Week 1	Week 2	Week 3	Week 4
Real	6.67	5.45	7.29	5.96
SmartSPEC	8.63	10.0	7.16	8.61
RAND	14.20	13.92	14.01	13.65
BROW	12.29	12.37	12.75	12.34
LÉVY	13.83	13.49	13.64	13.23
EPR	14.75	12.86	14.83	10.05

Occupancy Difference

- On average, there was a **35% difference in trajectory distances** between SmartSPEC and the campus dataset
- On average, there was a **36% difference in occupancy counts per space** between SmartSPEC and the campus dataset.
- Most mobility models do significantly worse.

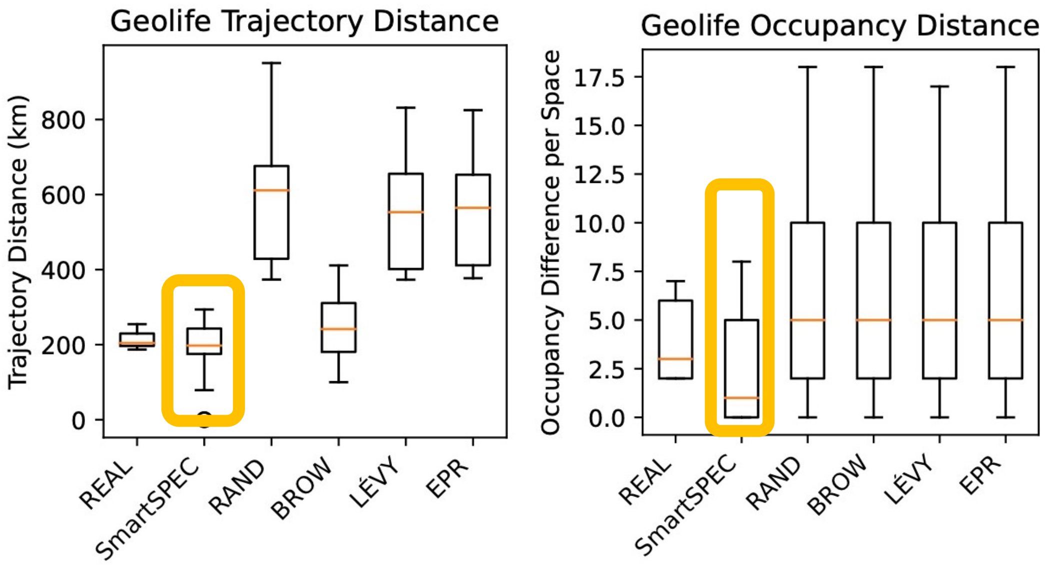
SmartSPEC produces trajectories and occupancy counts that are close to real data on the scope of a campus building

Evaluating Realism in City Scenario

- On average, there was a **13% difference in trajectory distances** between SmartSPEC and the GeoLife dataset
- On average, there was a **37% difference in occupancy counts per space** between SmartSPEC and the GeoLife dataset.
- Brownian motion baseline creates similar trajectories to real data, but have very different occupancy

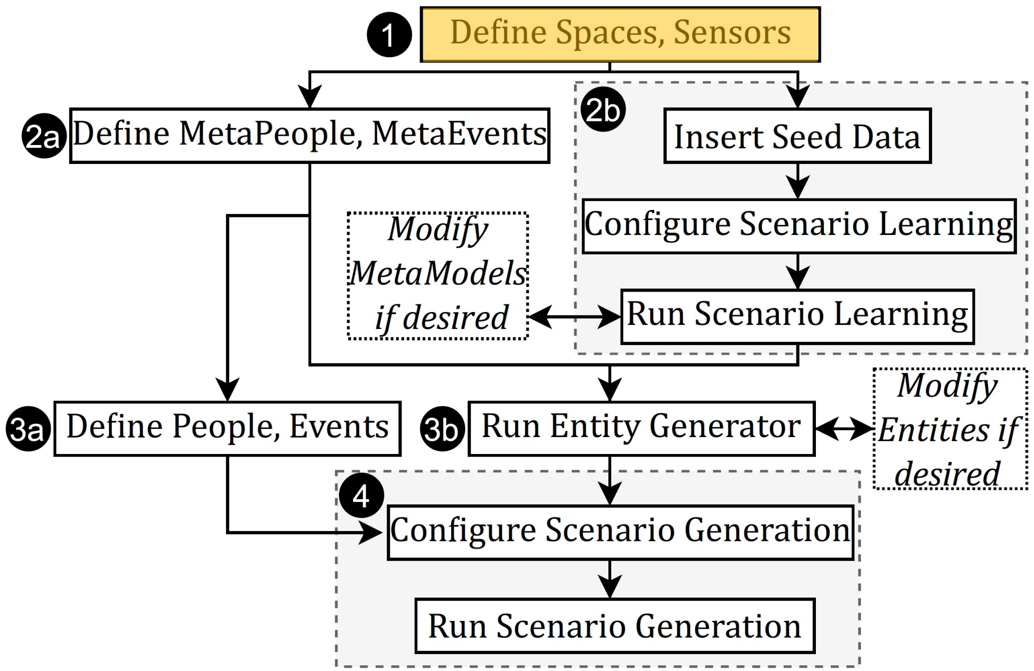
SmartSPEC produces trajectories and occupancy counts that are close to real data on the scope of a city

City Scenario



Trajectory Distance (km) Occupancy Distance

SmartSPEC: Workflow




```
[{"id": 1,
  "description": "lobby",
  "coordinates": [30,50,10],
  "capacity": 30,
  "neighbors": [2,3] }, ...]
```

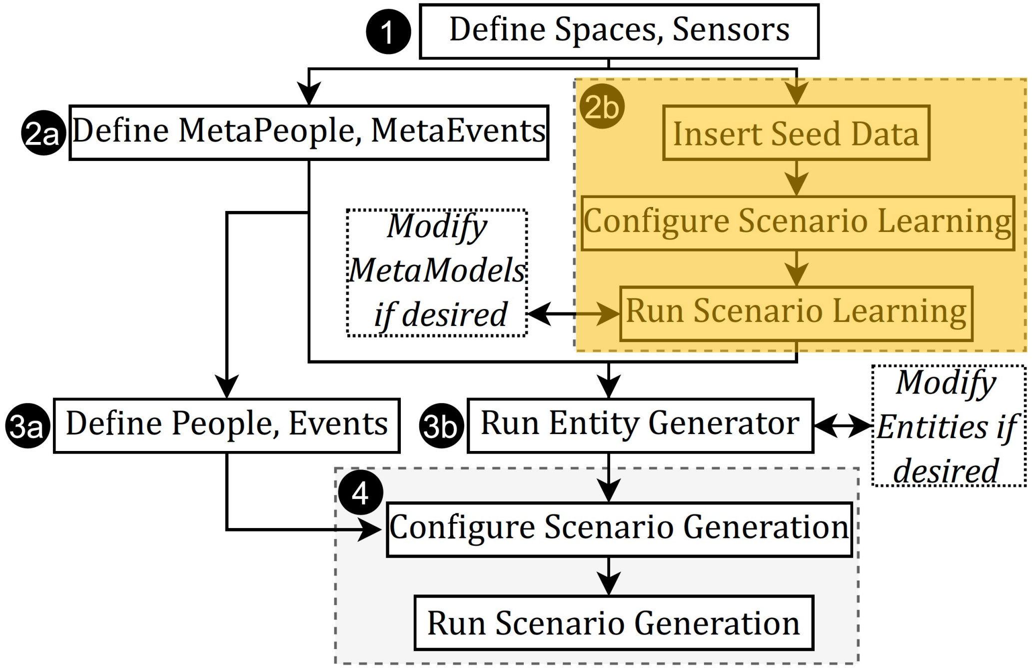

Sample Space File


```
[{"id": 3,
  "description": "AP-2081"
  "mobility": "static",
  "coverage": [1,3],
  "interval": 60 }, ...]
```


Sample Sensor File

Our code is publicly available on GitHub: https://github.com/andrewgchio/SmartSPEC

SmartSPEC: Workflow




```
wifi_ap,cnx_time,client_id
1,2017-04-09 07:30:31,81
9,2017-04-09 10:39:13,72
8,2017-04-09 10:40:08,72
...
```

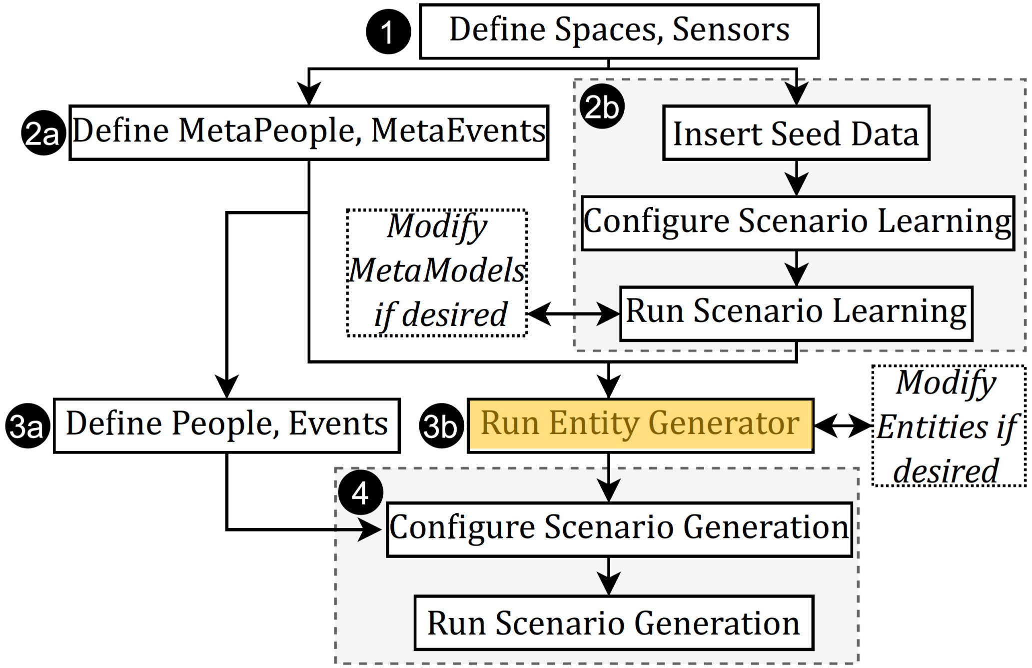
Sample Seed Data

```
[learners]
start       = 2017-04-01
end         = 2017-05-01
unit        = 5
validity    = 10
smooth      = EMA
window      = 10
time-thresh = 30
occ-thresh  = 1

[filepaths]
spaces     = data/demo/Spaces.json
sensors    = data/demo/Sensors.json
metaevents = data/demo/MetaEvents.json
metapeople = data/demo/MetaPeople.json
...
```

Sample Configuration File for Scenario Learning

Our code is publicly available on GitHub: https://github.com/andrewgchio/SmartSPEC

SmartSPEC: Workflow

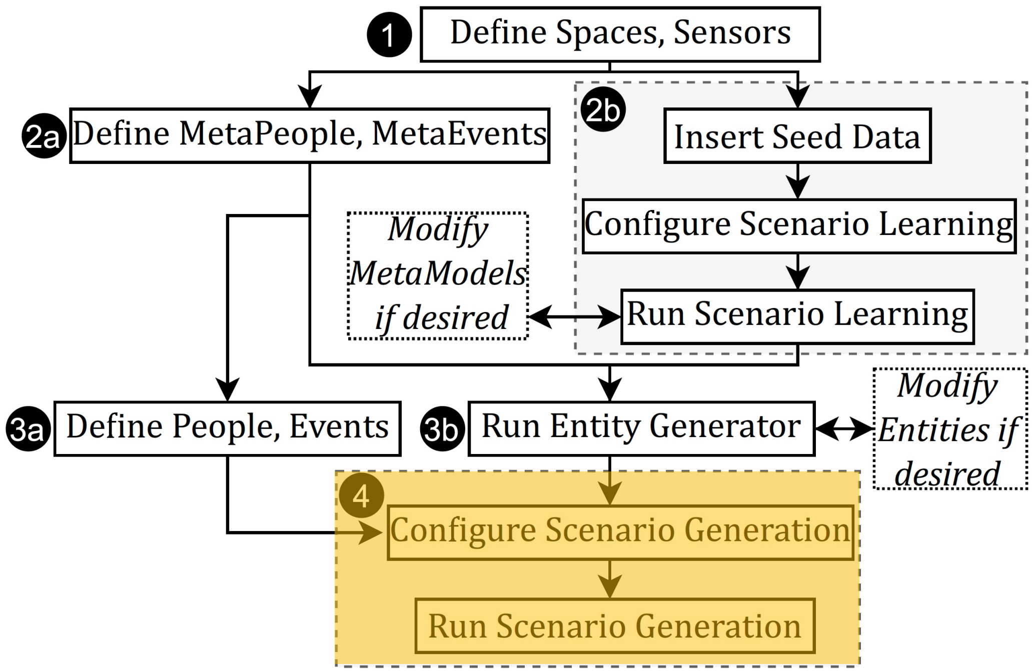



```
[people]
number = 500
generation = all

[events]
number = 5000
generation = diff

[synthetic-data-generator]
start = 2018-01-08
end   = 2018-01-29

[filepaths]
metapeople   = data/demo/MetaPeople.json
metaevents   = data/demo/MetaEvents.json
spaces       = data/demo/Spaces.json
sensors      = data/demo/Sensors.json
people       = data/demo/People.json
events       = data/demo/Events.json
output       = data/demo/output/
...
```

Sample Configuration File for Scenario Generation

Our code is publicly available on GitHub: https://github.com/andrewgchio/SmartSPEC

SmartSPEC: Workflow




```
[people]
number = 500
generation = all

[events]
number = 5000
generation = diff

[synthetic-data-generator]
start = 2018-01-08
end   = 2018-01-29

[filepaths]
metapeople   = data/demo/MetaPeople.json
metaevents   = data/demo/MetaEvents.json
spaces       = data/demo/Spaces.json
sensors      = data/demo/Sensors.json
people       = data/demo/People.json
events       = data/demo/Events.json
output       = data/demo/output/
...
```

Sample Configuration File for Scenario Generation

Our code is publicly available on GitHub: https://github.com/andrewgchio/SmartSPEC

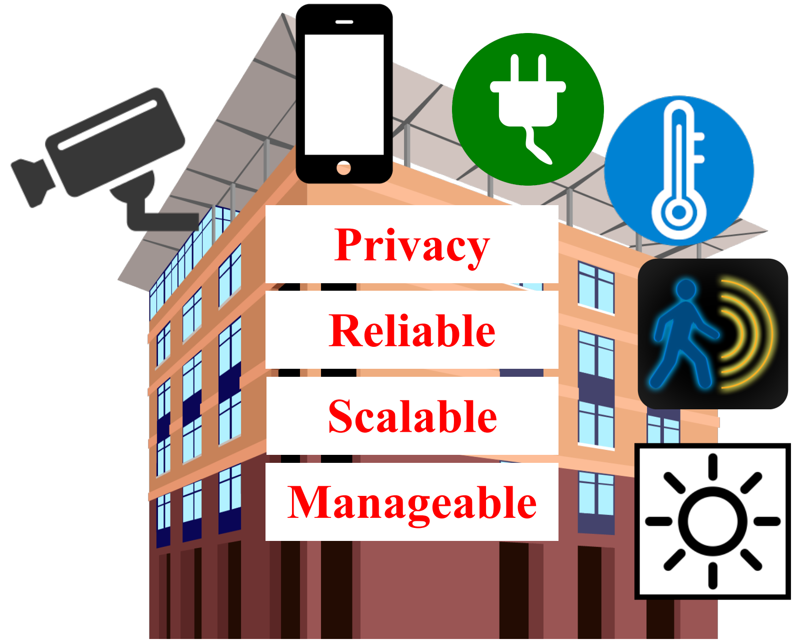
SmartSPEC: Applicability and Utility

```
PersonID,EventID,SpaceID,StartDatetime,EndDatetime
17,2698,1100,2018-01-15 09:51:50,2018-01-15 09:54:20
33,4200,1422,2018-01-15 09:59:55,2018-01-15 10:46:04
42,613,1420,2018-01-15 09:57:27,2018-01-15 10:44:10
60,1660,1422,2018-01-15 09:59:19,2018-01-15 10:37:00
71,401,1433,2018-01-15 09:59:55,2018-01-15 10:44:30
95,3609,1425,2018-01-15 09:58:32,2018-01-15 10:46:58
134,4200,1422,2018-01-15 09:58:26,2018-01-15 10:41:59
134,0,1100,2018-01-15 09:46:19,2018-01-15 09:48:21
166,1015,1300,2018-01-15 09:59:55,2018-01-15 10:47:16
175,1038,1200,2018-01-15 09:46:53,2018-01-15 09:49:37
177,3335,1422,2018-01-15 09:56:56,2018-01-15 10:41:38
...
```

Sample of Synthetic Data Output

Sample Generated Dataset

SmartSPEC: Applicability and Utility

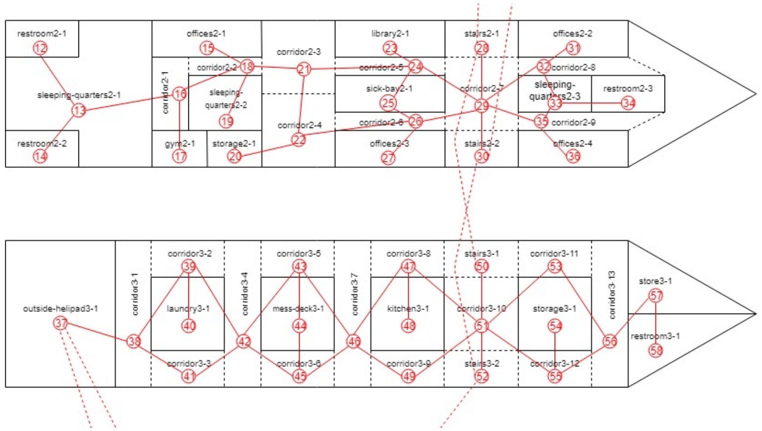


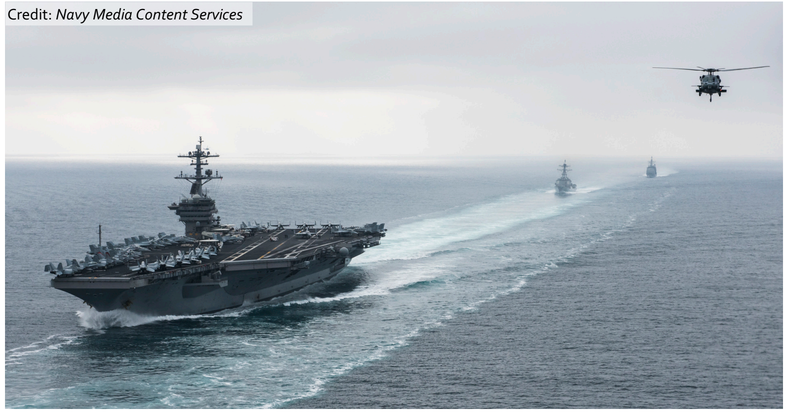
TIPPERS: Testbed for IoT-based Privacy-Preserving PERvasive Spaces

- Design robust, experimental testbed
- Explore privacy technologies
- Real-world deployments

NAVWAR Trident Warrior:

- Explore potential benefits of IoT technologies for naval use cases
- Day in the life of a sailor in mission-critical scenarios and non-mission-critical scenarios
  - Simulated activities on a Navy Ship




Credit: *Navy Media Content Services*

Key Takeaways

- **Realistic and Semantically Explainable data** are required to test and validate smart space approaches

- We developed SmartSPEC: an **event-driven** smart space simulator
  - Customizable smart space datasets using models of entities in smart space ecosystems.
  - ML techniques to learn profiles of people and types of events from seed data

- We presented a **structured methodology to evaluate the realism of synthetic data.**

- Our experiments show that SmartSPEC produces data that is **1.4x -4.4x** more realistic than baselines.

- The SmartSPEC approach can also be employed to generate synthetic sensor data.

- **Our code is publicly available on GitHub:** https://github.com/andrewgchio/SmartSPEC