III. CHITCHAT CONTEXT SUMMARIES

A *context summary*'s attributes describe an entity. We provide a set of scenarios in which context plays a pivotal role and define data types common to context values. We then introduce CHITCHAT's *filters*, which allow a context summary's contents (and the recipient's own situation) to determine whether values in the summary are likely false positives. Our prior work [9] provided an API for applications to specify these filters. In this paper, we apply the filters to our new context structures to help preserve the semantic value of the information they contain. These filters allow us to use structures that have theoretically high false positive rates that can be reduced in practice.

A. Scenarios

One category of scenarios for which D2D sharing is particularly applicable is when communication infrastructures are unusable. The goal is to collect and share context about individuals and the environment; the information could be critical in analyzing conditions to form an evacuation plan.
An earthquake has caused local blackouts and dangerous situations, trapping people and leaving the communication infrastructures inoperable. An individual's context is:

```
{ "latitude": [30, 25, 38, 2],
  "longitude": [-17, 47, 11, 0],
  "time": [11, 21],
  "date":[2015, 10, 11],
  "age": 23,
  "name": "Brian Taylor",
  "temperature": 41,
  "temperature unit": "C",
  "message": "Trapped in room 121" }
```

Similarly, information from structural sensors that survived the earthquake can share information about the building:

```
{ "latitude": [30, 25, 38, 5],
  "longitude": [-17, 47, 11, 0],
  "time": [11, 21],
  "date":[2015, 10, 11],
  "device id": 11,
  "number of sensor": 3,
  "sensor 1 name": "temperature",
  "sensor 1 value": 28,
  "sensor 1 unit": "C",
  "sensor 2 name": "humidity",
  "sensor 2 value": 43,
  "sensor 2 unit": "%",
  "sensor 3 name": "light",
  "sensor 3 value": 121,
  "sensor 3 unit": "lux" }
```

Alternatively, a device could process the data minimally, e.g., the three `sensor 1` attributes could be replaced by two:

```
  ...
  "temperature": 28,
  "temperature unit": "C"
  ...
```

In our evaluation, we use the first summary because it is indicative of the summary needed when no processing is done by the microcontroller collecting the sensor value.

Another motivating scenario is hyper-localized search, with the goal of sharing information about the environment and individuals' interests to help direct the individuals.

A book lover visits an open-air book fair. Her device notes and shares her interest in modern art books. A dynamic social network formed from booksellers and other visitors with similar interests can provide a small network from which to get hints of the location of the best deals.

```
{ "latitude": [31, 25, 38, 2],
  "longitude": [-17, 42, 11, 0],
  "date":[2015, 10, 09],
  "time": [10, 21],
  "leave time": [12, 21],
  "gchat id": "mary.zwky",
  "interest category 1": "art books",
  "interest item 1": "20 century European painting",
  "interest category 2": "paper",
  "interest item 2": "Roylco R15286 Antique Paper",
  "special interest item": "Hand made notebooks" }
```

The visitor's summary could be propagated until it reaches an art bookseller who could follow up via gchat. Alternatively, the context could be compared to that of similar visitors, bootstrapping sharing other context, e.g., sellers' service quality.

CHITCHAT applies generally to situations that involve contexts that exhibit the following characteristics:

- Context information is hyper-localized in space and time.
- Context attributes are related to each other.
- Strings are frequently used to describe context.
- Many context values need not be extremely accurate.
- Context values often have range limits.

We next examine these types more closely, looking at representations best suited to compact context summaries.

B. CHITCHAT *Types*

CHITCHAT defines data types tailored to pervasive computing context; specific examples are shown in Table I. We support multiple integral types and assume only single precision (32 bit) floating point numbers (a higher precision is not commonly required for context). We use Pascal-style strings, with the length as the first element. We also define special types to aid in efficiently packing data, e.g., a "level" type that scales from 1 and 10. Dates include a year (7 bits), month (4 bits), and day (5 bits) and times have hours (5 bits) and minutes (6 bits)[1]. Both latitude and longitude are expressed in degrees (± 90 for latitude and ± 180 for longitude), minutes, seconds, and subseconds. This gives sufficient precision ($\sim \pm 30cm$), given that the precision of a typical GPS unit is $\sim 1 - 10m$.

TABLE I: Example data types for contexts

Type	Bits	Bytes	Range	Encoding
Boolean	1	1	(0,1)	
Unsigned byte	8	1	(0, 255)	
Byte	8	1	(-128, 127)	
Float	32	4		IEEE 754
String	$n \times 8$	n		Pascal
Age	7	1	(0, 127)	
Level	4	1	(1, 10)	
Date	16	2		(7,4,5)
Time	11	2		(5,6)
Latitude	27	4		(8,6,6,7)
Longitude	28	4		(9,6,6,7)

[1]Applications requiring second granularity can easily add another integer value; we simply do not support seconds in the default time type.