‘Networks of Things’

The Pieces, Parts, and Data

J. Voas

j.voas@ieee.org

Primitives

1. **Sensor** A *sensor* is an electronic utility that digitally measures physical properties such as temperature, acceleration, weight, sound, etc.
2. **Aggregator** An *aggregator* is a software implementation based on mathematical function(s) that transforms groups of raw data into *intermediate* data.
3. **Communication channel** A *communication channel* is a medium by which data is transmitted (e.g., physical via USB, wireless, wired, verbal, etc.).
4. ***e*Utility** A *e*Utility (external utility*)* is a software or hardware product or service.
5. **Decision trigger** A *decision* trigger creates the final result(s) needed to satisfy the purpose, specification, and requirements of a specific NoT.

Sensor

1. Sensors are physical.
2. Sensors may have little or no software functionality and computing power; more advanced sensors may have software functionality and computing power.
3. Sensors will likely be heterogeneous, from different manufacturers, and collect data, with varying levels of data integrity.
4. Sensors will have operating geographic locations that may change.
5. Sensors may provide surveillance. Cameras and microphones are sensors.
6. Sensors may have an owner(s) who will have control of the data their sensors collect, who is allowed to access it, and when.
7. Sensors will have pedigree – geographic locations of origin and manufacturers. Pedigree may be unknown and suspicious.
8. Sensors may fail continuously or fail intermittently.
9. Sensors may be cheap, disposable, and susceptible to wear-out over time; here, building security into a specific sensor will rarely be cost effective. However there will differentials in security, safety, and reliability between consumer grade, military grade, industrial grade, etc.

Sensor

10. Sensors may return no data, totally flawed data, partially flawed data, or correct/acceptable data.
11. Sensors are expected to return data in certain ranges, e.g., [1 … 100]. When ranges are violated, rules may be needed on whether to turn control over to a human or machine when ignoring out-of-bounds data is inappropriate.
12. Sensor repair is likely handled by replacement.
13. Sensors may be acquired off-the-shelf.
14. Sensors release data that is event-driven, driven by manual input, or released at pre-defined times.
15. Sensors may have a level of data integrity ascribed (Weights).
16. Sensors may have their data encrypted to void some security concerns Sensor data may be leased to multiple NoTs. A sensor may have multiple recipients of its data.
17. The frequency with which sensors release data impacts the data's currency and relevance. Sensors may return valid data at an incorrect rate/speed.
18. Sensor data may be 'at rest' for long periods of time; sensor data may become *stale.*
19. A sensor's resolution may determine how much information is provided.
20. Security is a concern for sensors if they or their data is tampered with or stolen.
21. Reliability is a concern for sensors.

Aggregator

1. Aggregators are likely virtual due the benefit of changing implementations quickly and increased malleability. A situation may exist where aggregators are physically manufactured, e.g., a FPGA or hard-coded aggregator that is not programmable, similar to an *n*-version voter.
2. Aggregators are assumed to lack computing horsepower, however this assumption can be relaxed by changing the definition and assumption of virtual to physical, e.g. firmware, microcontroller or microprocessor. Aggregators will likely use weights to compute intermediate data.
3. Aggregators have two actors that make them ideal for consolidating large volumes of data into lesser amounts: Clusters, and Weights. Aggregator is the *big data processor* within IoT.
4. Intermediate data may suffer from some level of *information loss*.
5. For each cluster there should be an aggregator or set of potential aggregators.
6. Aggregators are executed at a specific time and for a fixed time interval.
7. Aggregators may be acquired off-the-shelf.
8. Security is a concern for aggregators (malware or general defects) and for the sensitivity of their aggregated data.
9. Reliability is a concern for aggregators (general defects).

Actor: Cluster

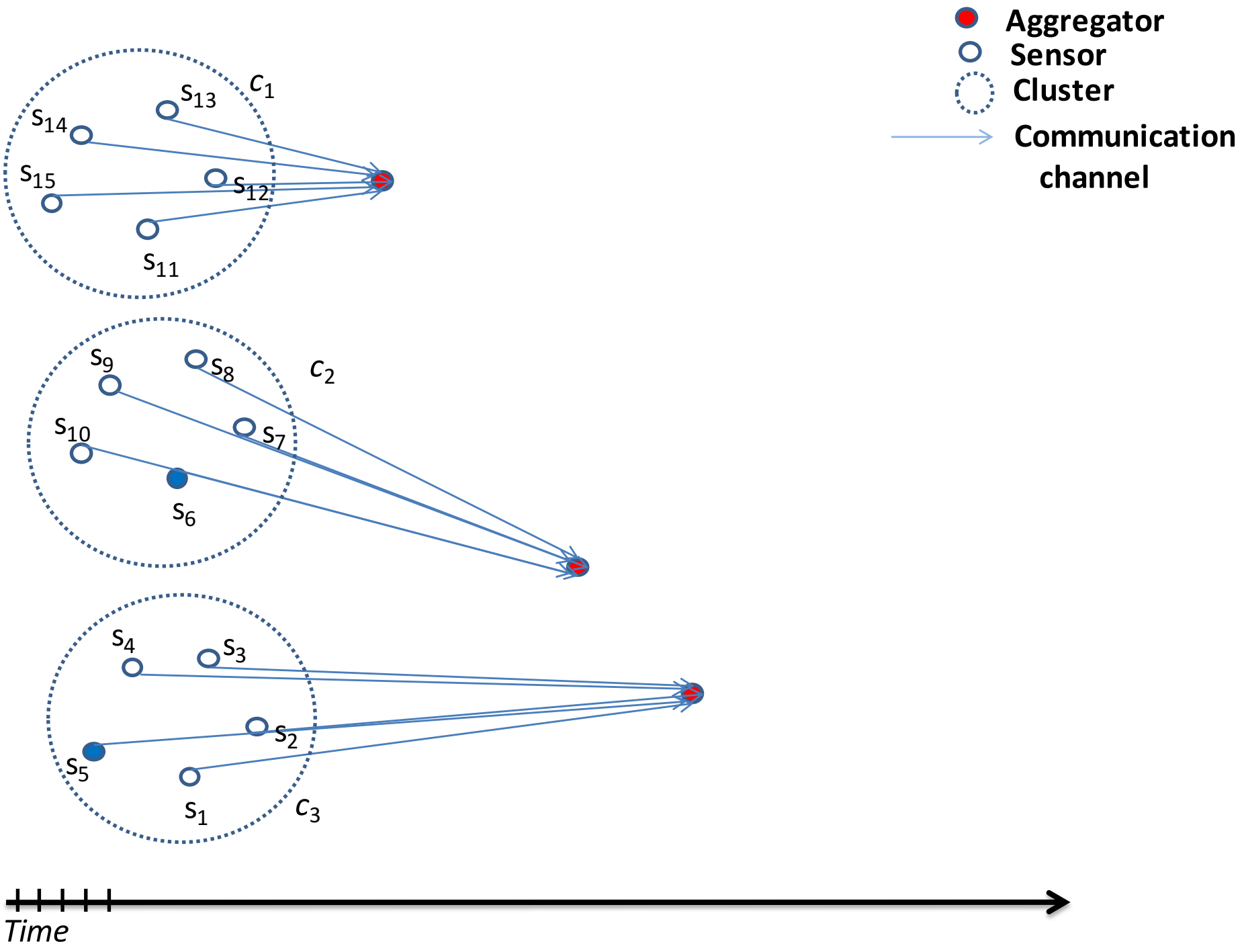
1. Clusters are abstractions of a set of sensors along with the data they output—clusters may be created in an *ad hoc* manner or organized according to fixed rules.
2. Clusters are not inherently physical.
3. C_i is essentially a *cluster* of the sensor data from $n \geq 1$ sensors, $\{d_1, d_2, d_3, \ldots, d_n\}$.
4. C_i may share one or more sensors with C_k, where $i \neq k$.
5. *Continuous-binding* of a sensor to a cluster may result in little ability to mitigate trustworthiness concerns if the binding is *late*.
6. Clusters are malleable and can change their collection of sensors and their data.
7. How clusters are composed is dependent on what mechanism is employed to aggregate the data, which ultimately impacts the purpose and requirements of a specific NoT.

Actor: Weight

1. A weight may be hardwired or modified on the fly.
2. A weight may be based on a sensor's perceived trustworthiness, e.g., based on who is the sensor's owner, manufacturer, geographic location of manufacture, geographic location where the sensor is operating, sensor age or version, previous failures or partial failures of sensor, sensor tampering, sensor delays in returning data, etc. A weight may also be based on the value of the data, uniqueness, relation to mission goals, etc.
3. Different NoTs may leverage the same sensor data and re-calibrate the weights per the purpose of a specific NoT.
4. Aggregators may employ artificial intelligence to modify their clusters and weights.
5. Weights will affect the degree of information loss during the creation of intermediate data.
6. Security is probably not a concern for weights unless they are tampered with.
7. The appropriateness (or correctness) of the weights is crucial for the purpose of a NoT.

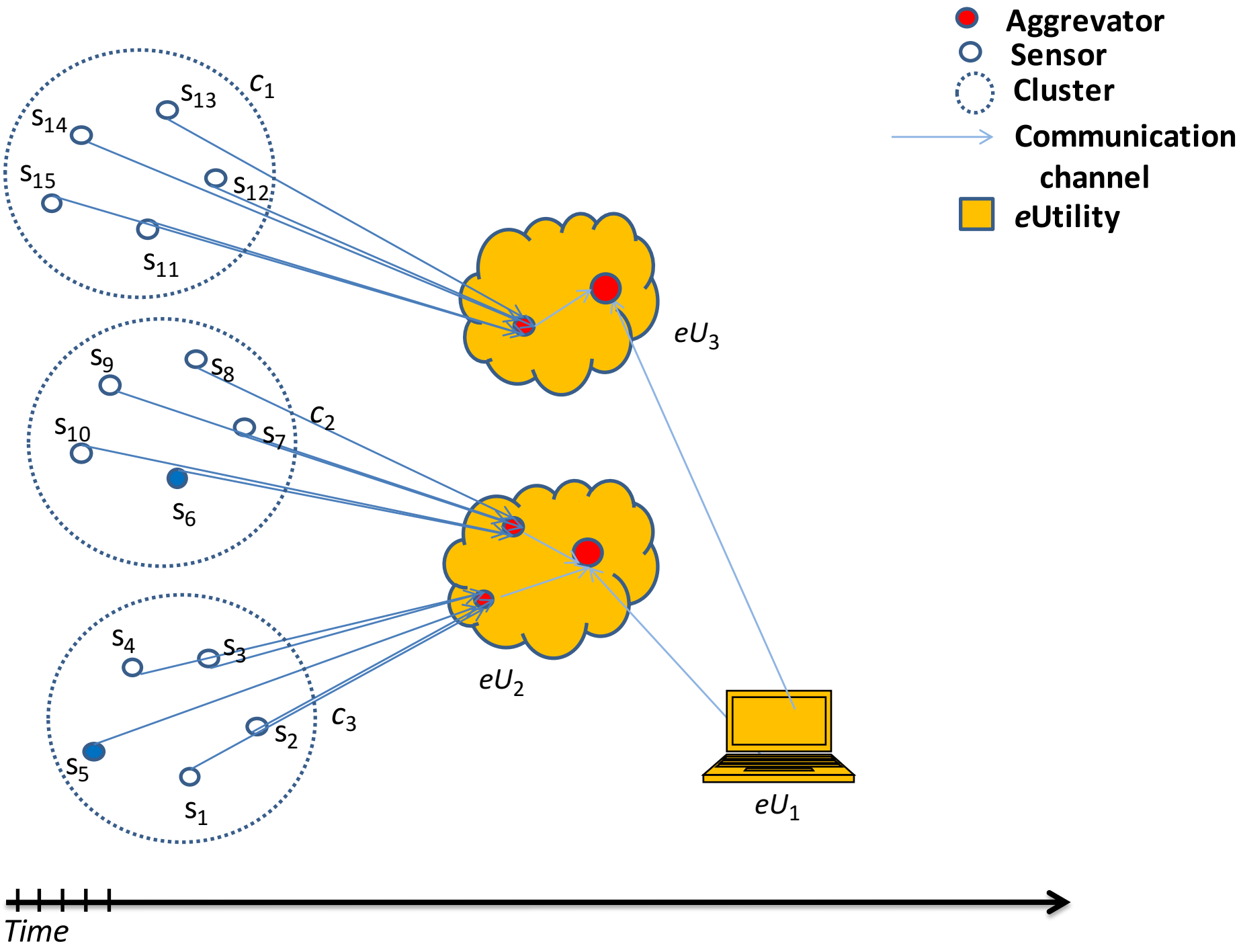
Communication channel

1. Communication channels move data between computing and sensing.
2. Since data is the “blood” of a NoT, communication channels are the “veins” and “arteries”.
3. Communication channels will have a physical or virtual aspect to them, or both. For example protocols and associated implementations provide a virtual dimension, cables provide a physical dimension.
4. Communication channel dataflow may be unidirectional or bi-directional. There are a number of conditions where an aggregator might query more advanced sensors, or potentially recalibrate them in some way (e.g., request more observations per time interval).
5. No standardized communication channel protocol is assumed; a specific NoT may have multiple communication protocols between different entities.
6. Communication channels may be wire-less.
7. Communication channels may be an offering (*service* or *product*) from 3rd party vendors.
8. Communication channel *trustworthiness* may make sensors appear to be failing when actually the communication channel is failing.
9. Communication channels are prone to disturbances and interruptions.
10. *Redundancy* can improve communication channel reliability.
11. Performance and availability of communication channels will greatly impact any NoT that has time-to-decision requirements (the Decision trigger primitive is discussed later).
12. Security and reliability are concerns for communication channels.


Aggregator
Sensor
Cluster
Communication channel
c_1
s_{13}
s_{14}
s_{15}
s_{12}
s_{11}
s_9
s_8
c_2
s_{10}
s_7
s_6
s_4
s_3
s_2
s_5
s_1
c_3
Time

*e*Utility

1. *e*Utilities execute processes or feed data into the overall dataflow of a NoT.
2. *e*Utilities may be acquired off-the-shelf.
3. *e*Utilities include databases, mobile devices, misc. software or hardware systems, clouds, computers, CPUs, actuators, etc. The *e*Utility primitive can be subdivided.
4. *e*Utilities, such as clouds, provide computing power that aggregators may not have.
5. A human may be viewed as a *e*Utility.
6. Data supplied by a *e*Utility may be weighted.
7. An *e*Utility may be counterfeit; this is mentioned later in element Device_ID (Section 3).
8. Non-human *e*Utilities may have Device_IDs; Device_IDs may be crucial for authentication.
9. Security and reliability are concerns for *e*Utilities.

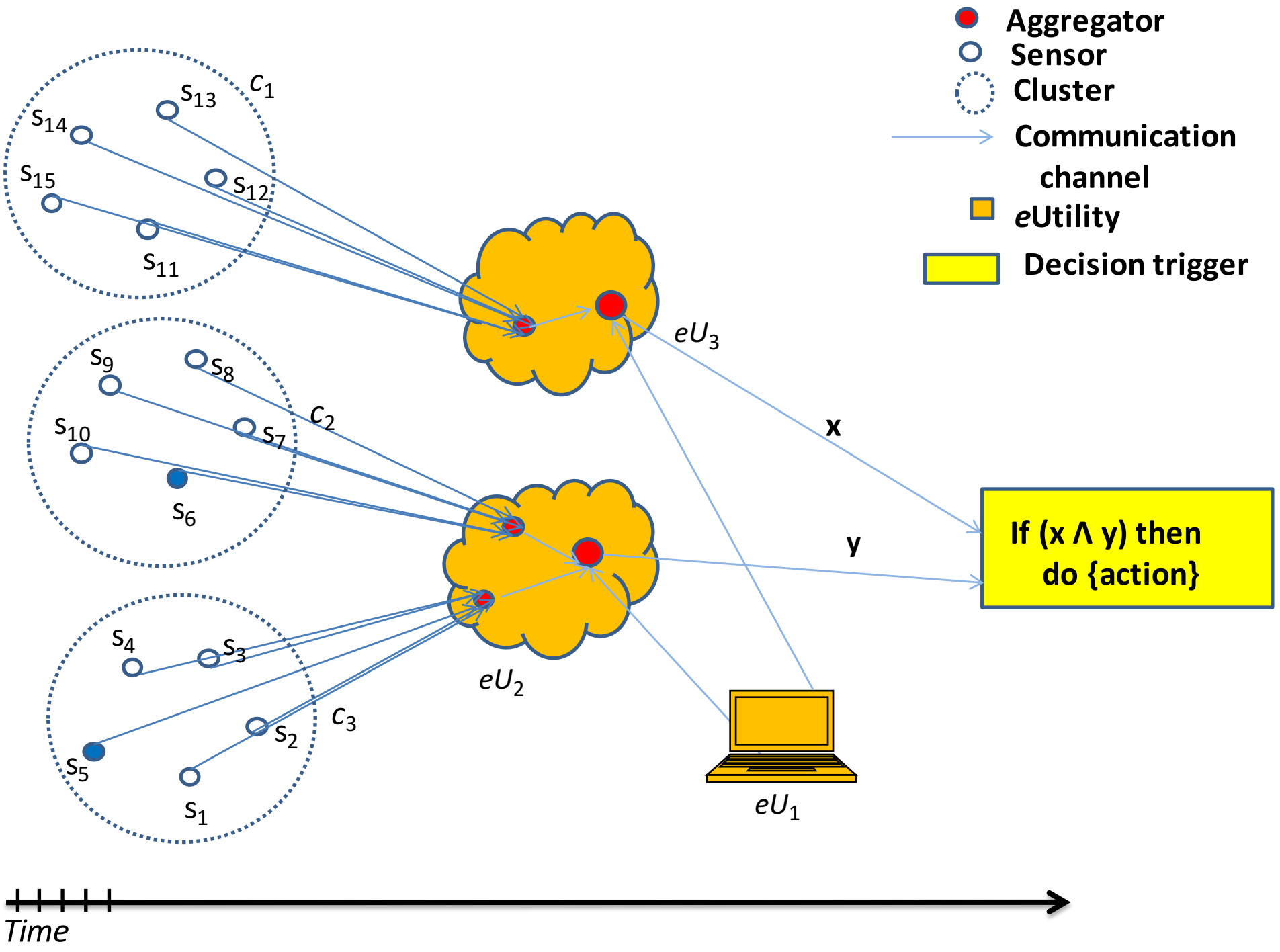

Aggrevator
Sensor
Cluster
Communication channel
eUtility
c_1
s_{13}
s_{14}
s_{15}
s_{12}
s_{11}
eU_3
s_9
s_8
c_2
s_{10}
s_7
s_6
eU_2
s_4
s_3
c_3
s_2
s_5
s_1
eU_1
Time

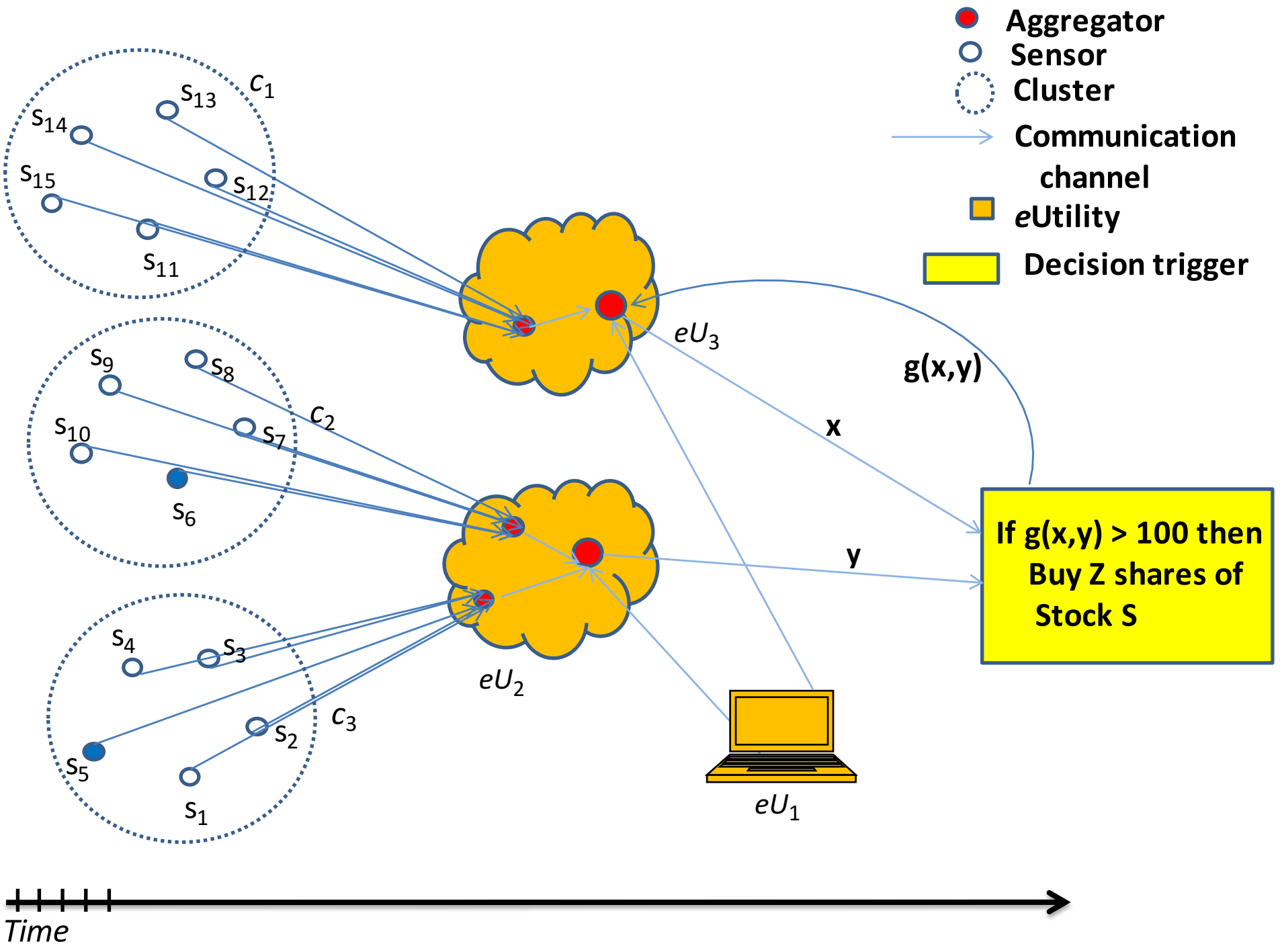
Decision trigger

1. A decision trigger is a pre-condition that must be TRUE before a NoT takes action.
2. A decision trigger should have a corresponding virtual implementation.
3. A decision trigger may have a unique owner.
4. Decision triggers may be acquired off-the-shelf or homegrown.
5. Decision triggers are executed at specific times and may occur continuously as new data becomes available.
6. Decision trigger results may be predictions.
7. Decision trigger results may control actuators or other transactions.
8. If a decision trigger feeds data signals into an actuator, then the actuator should be considered as a *e*Utility if the actuator feeds data back into the NoT.

Decision trigger

9. A decision trigger may feed its output back into the NoT creating a feedback loop.
10. It is fair to view a decision trigger as an **if-then** rule, although they will not all have this form.
11. The workflow up to decision trigger execution may be partially parallelizable.
12. Failure to execute decision triggers at time t_x may occur due to tardy data collection, inhibited sensors or *e*Utilities, inhibited communication channels, low performance aggregators, and a variety of other subsystem failure modes.
13. Economics and costs play a role in the quality of the decision trigger's output.
14. There may be intermediate decision triggers at any point in a NoT's workflow.
15. Decision triggers act similarly to aggregators and can be viewed as a special case of aggregator.


Aggregator
Sensor
Cluster
Communication channel
eUtility
Decision trigger
c1
s13
s14
s15
s12
s11
eU3
s9
s8
c2
s10
s7
s6
x
y
If (x Λ y) then do {action}
s4
s3
eU2
c3
s2
s5
s1
eU1
Time


Aggregator
Sensor
Cluster
Communication channel
eUtility
Decision trigger
c1
s13
s14
s15
s12
s11
c2
s9
s8
s10
s7
s6
c3
s4
s3
s2
s5
s1
eU3
eU2
eU1
x
y
g(x,y)
If g(x,y) > 100 then Buy Z shares of Stock S
Time

Elements

1. **Environment** – The universe that all primitives in a specific NoT operate in; this is essentially the *operational profile* of a NoT. An analogy is the various weather profiles that an aircraft operates in or a particular factory setting that a NoT operates in. This will likely be very difficult to correctly define.
2. **Cost** – The expenses, in terms of time and money, that a specific NoT incurs in terms of the non-mitigated reliability and security risks; additionally, the costs associated with each of the primitive components needed to build a NoT. Cost is an estimation or prediction. Cost drives the design decisions in building a NoT.
3. **Geographic location –** Physical place where a sensor or *e*Utility operates or was manufactured. Manufacturing location is a supply chain trust issue. Note that the operating location may change over time. Note that a sensor's or *e*Utility's geographic location along with communication channel reliability may affect the dataflow throughout the workflow in a timely manner. Geographic location determination may sometimes not be possible.
4. **Owner** - Person or Organization that owns a particular sensor, communication channel, aggregator, decision trigger, or *e*Utility. There can be multiple owners for any of these five. Note that owners may have nefarious intentions that affect overall trust. Note further that owners may remain anonymous.
5. **Device_ID** – A unique identifier for a particular sensor, communication channel, aggregator, decision trigger, or *e*Utility. This will typically originate from the originator of the entity, but it could be modified or forged.
6. **Snapshot** – an instant in time.

Special Element: Snapshot

1. Because a NoT is a distributed system, different events, data transfers, and computations occur at different snapshots.
2. Snapshots may be aligned to a clock synchronized within their own network. A global clock may be too burdensome for sensor networks that operate in the wild. Others, however, argue in favor of a global clock [Li 2004]. We do not endorse either scheme.
3. NoTs may affect business performance – sensing, communicating, and computing can speed-up or slow-down a NoT's workflow and therefore affect the "perceived" performance of the environment it operates in or controls.
4. Snapshots maybe tampered with, making it unclear when events actually occurred, not by changing time (which is not possible), but by changing the snapshot at which an event in the workflow triggers, e.g., sticking in a **delay()** function call.
5. Reliability and performance of a NoT may be highly based on (d).

Trustworthiness

Primitive or Element	Attribute	Pedigree Risk?	Reliability Risk?	Security Risk?
Sensor	Physical	Y	Y	Y
Snapshot	Natural phenomenon	N/A	Y	?
Aggregator	Virtual	Y	Y	Y
Communication channel	Virtual and/or Physical	Y	Y	Y
eUtility	Virtual or Physical	Y	Y	Y
Decision trigger	Virtual	Y	Y	Y
Geographic location	Physical (possibly unknown)	N/A	?	?
Owner	Physical (possibly unknown)	?	N/A	?
Environment	Virtual or Physical (possibly unknown)	N/A	Y	Y
Cost	Partially known	N/A	?	?
Device_ID	Virtual	Y	?	Y

Summary

1. It is unlikely a single, usable definition of IoT can be created and agreed upon
2. A common vocabulary is useful to foster dialogue concerning IoT
3. NoTs are the likely means by which IoT will be delivered
4. 5 primitives that impact the trustworthiness of NoTs are proposed
5. 6 elements that impact the trustworthiness of NoTs are proposed
6. IoT has a *big data* component. The aggregator primitive addresses much of that that.
7. The goal is to someday build definitions of IoT, and better address this assertion:

   ***Trust* in some NoT *A,* at some snapshot *X,* is a function of NoT *A*'s assets ϵ {sensor(s), aggregator(s), communication channel(s), *e*Utility(s), decision trigger(s)} with respect to the members ϵ {geographic location, owner, environment, snapshot, cost, Device_IDs}, for each asset in the first set, when applicable.**

Further Reading

A paper with more information is available at:
https://dslsrv.gmu.edu/DecodingIoTv1.0.pdf