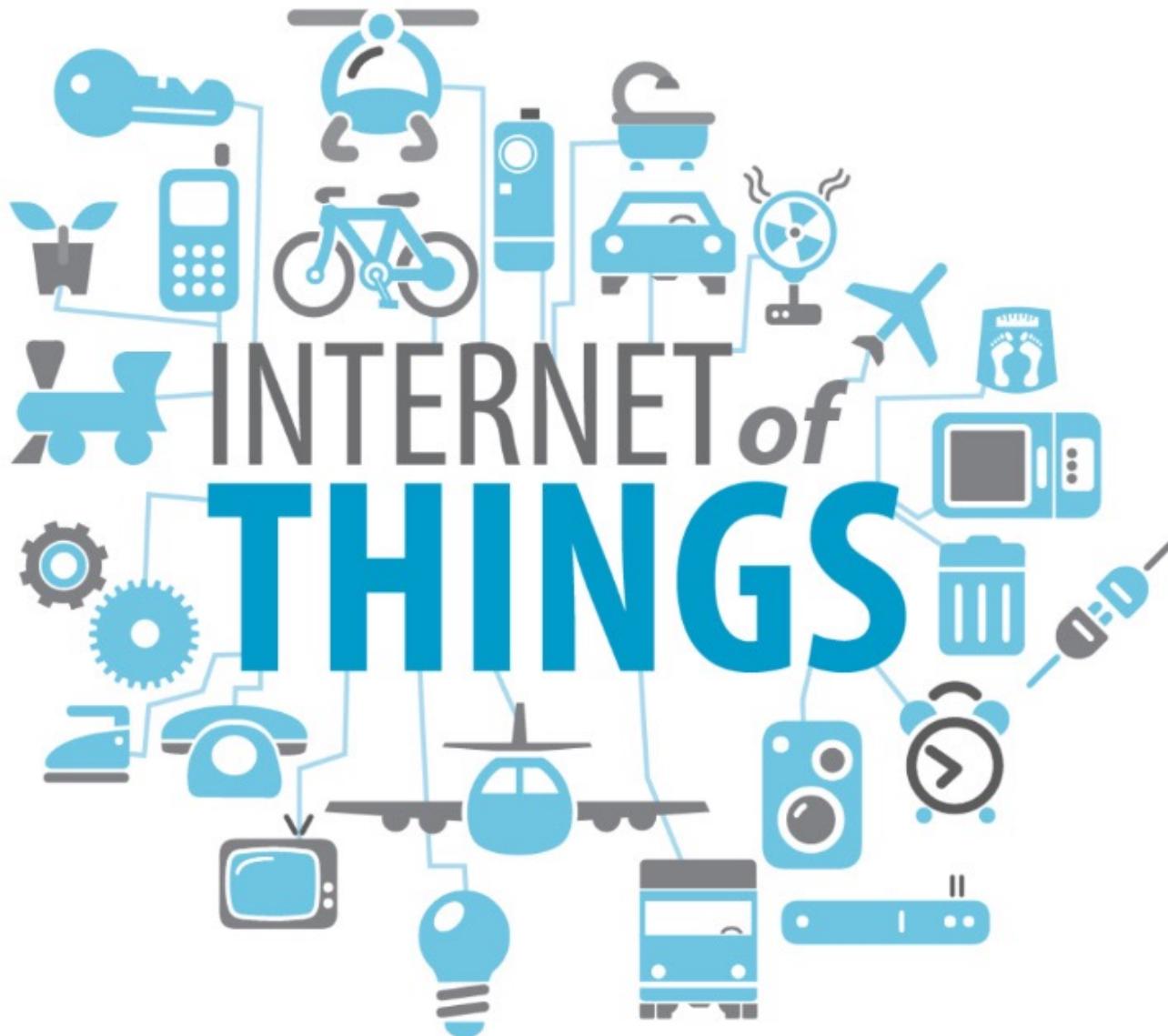


Tracking Sensitive and Untrustworthy Data in IoT

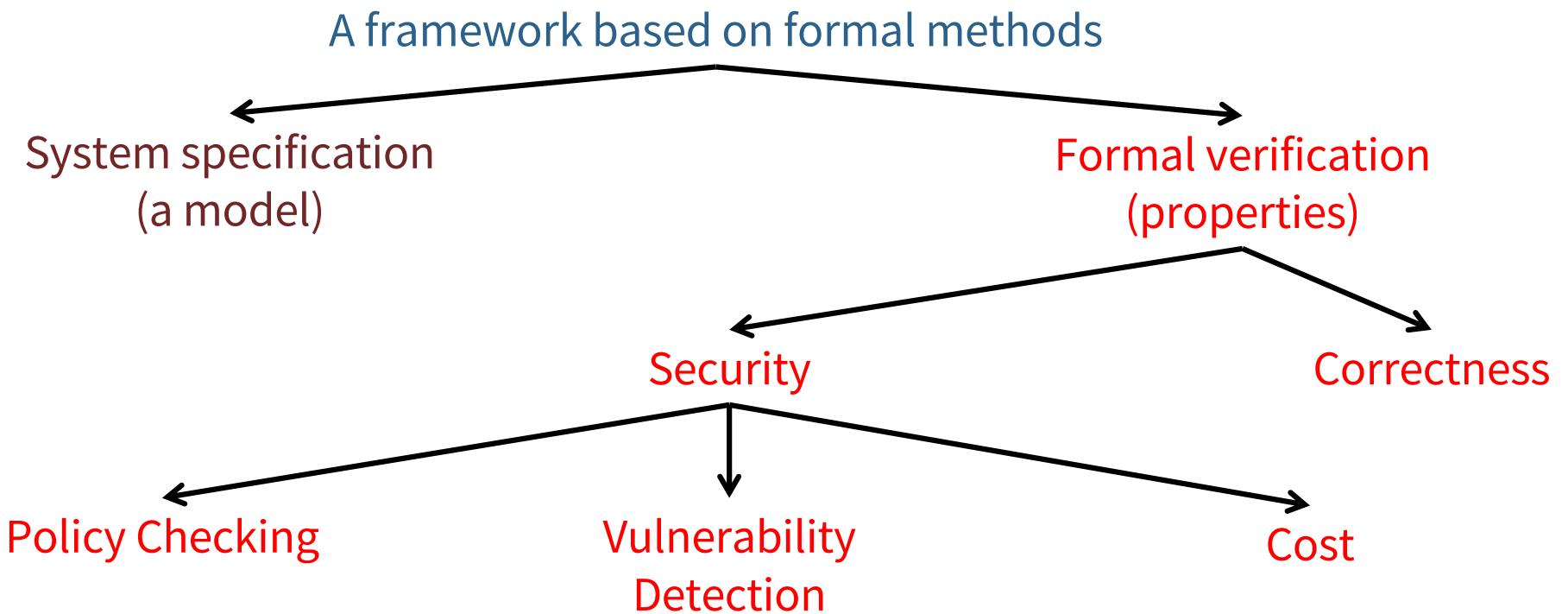
Chiara Bodei

Letterio Galletta

Adesso andiamo a vedere come si può fare il tracciamento di dati sotto forma di taint analysis, quindi l'idea è ancora quella che abbiamo portato avanti fino a ora io voglio avere un framework basato sui modelli formali che mi aiuti a specificare sistemi e allo stesso tempo a fare verifiche di proprietà sia dal punto di vista della correttezza della predizione che dal punto di vista della sicurezza, quindi in particolare posso andare a controllare le policy, controllare se ci sono vulnerabilità, poi questo non lo vedremo, ma anche posso andare anche a controllare i costi delle mie varie soluzioni. Quindi diciamo dovrebbe essere un modo per aiutare anche la security by design, cioè dovrei cercare di vedere a livello di modello astratto che cosa mi conviene fare per non avere problemi di sicurezza una volta implementato il modello.



Our long term goal



The goal: promoting a security by design methodology

A tool for reasoning abstractly about system design & detecting possible threats before deployment



Our proposal: IoT-LySa

A specification language for modelling system behaviour

Constructs

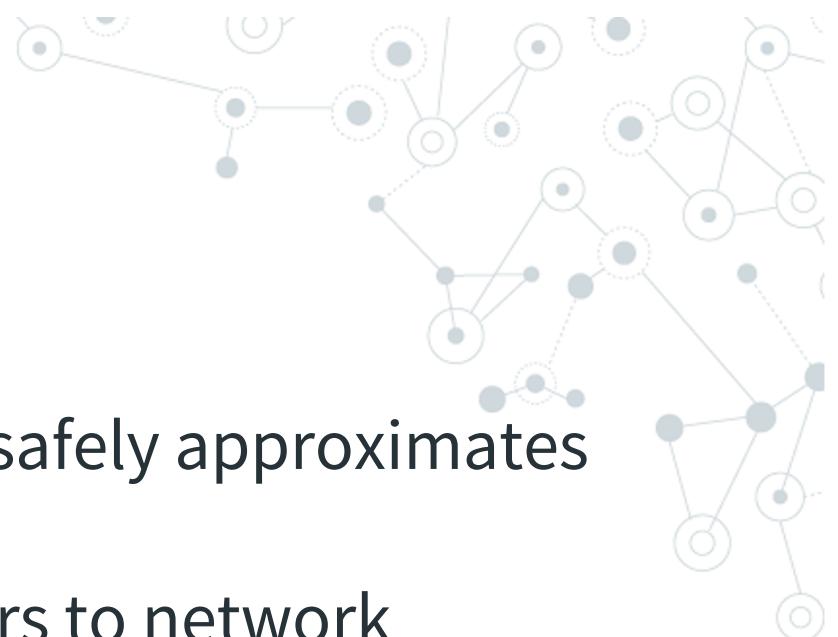
- Multiple nodes
- Sensors
- Actuators
- Group communication
- Cryptography



Perché si usa l'analisi statica, abbiamo detto proprio perché io voglio far capire cioè voglio aiutare i chi fa il progetto di una rete, di un sistema di nodi a ragionare su possibili flows che si possono verificare e anche scoprire possibili azioni a rischio in maniera che il design non possa essere avvisato e possa mettere in campo le opportune contromisure e poi è anche interessante dare un contributo dal punto di vista di un sorta di pratica che può essere quella del ragionamento di tipo what if, cioè cosa succede se questo sensore non va più bene cosa succede se questo nodo non è più affidabile? Quindi tutto il mio costrutto deve essere rivisto sotto questo aspetto, quindi io traccio tutto il comportamento ti dico che la decisione di attuazione di accendi la luce dipende da tutta una serie di parametri e mi dovrebbero portare alla decisione giusta? Ma cosa succede se su questi parametri qualcuno ha messo lo zampino e quindi mi porta a fare la scelta sbagliata.



IoT-LySa static analysis



Control Flow Analysis (CFA) to safely approximates

- ◎ Interaction among nodes
- ◎ How data spread from sensors to network
- ◎ How data are manipulated

Security checks based on analysis results

- ◎ Preventing leakage
- ◎ No read up/ no write down
- ◎ Selective data propagation policy
- ◎ Taint analysis



Why static analysis?

- Help designers to reason about possible flaws in the early stages of development
- Detect possible risky actions
- Give designers hints about possible countermeasures to adopt at runtime

Example: What happens if an attacker can tamper with a sensor? (“What if” reasoning)

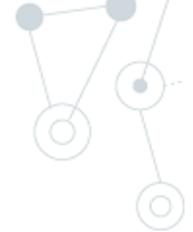


Static taint analysis

Quindi l'idea è cercare di capire quanto un design è abbastanza robusto rispetto alle manipolazioni dei dati e capire soprattutto se ci sono delle computazioni particolarmente critiche che possono dipendere da dati che possono essere in qualche maniera manomessi. Quello che facciamo, naturalmente, basandosi sul nostro solito framework è fare l'albero over approximation del comportamento e vedere come è possibile tracciare dati da questo punto di vista.

Goal: Determine whether our design is robust against data manipulations

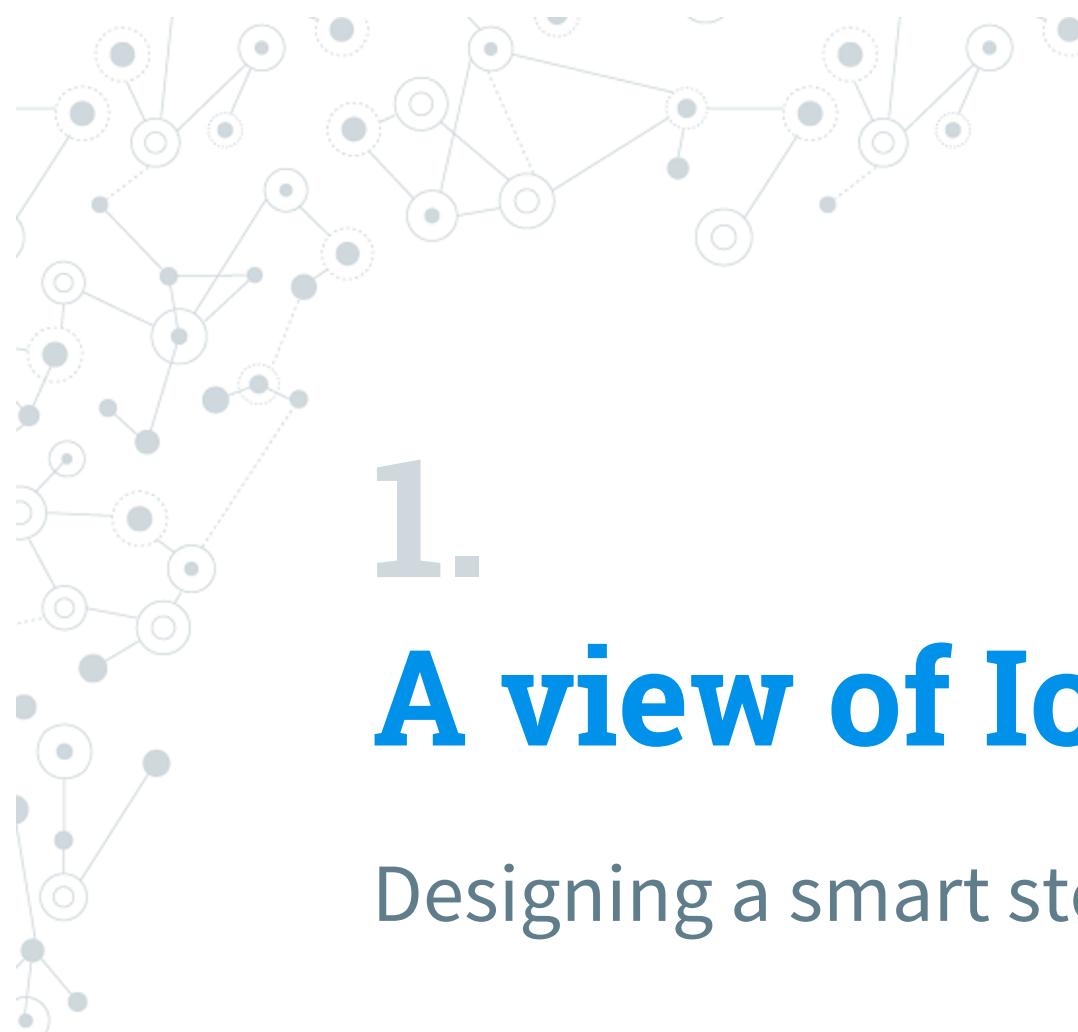
- Do critical computations depend on manipulated data?



How: Over-approximate from which data sources computations depend

- Do they depend on sensitive or possible tamperable data sources?





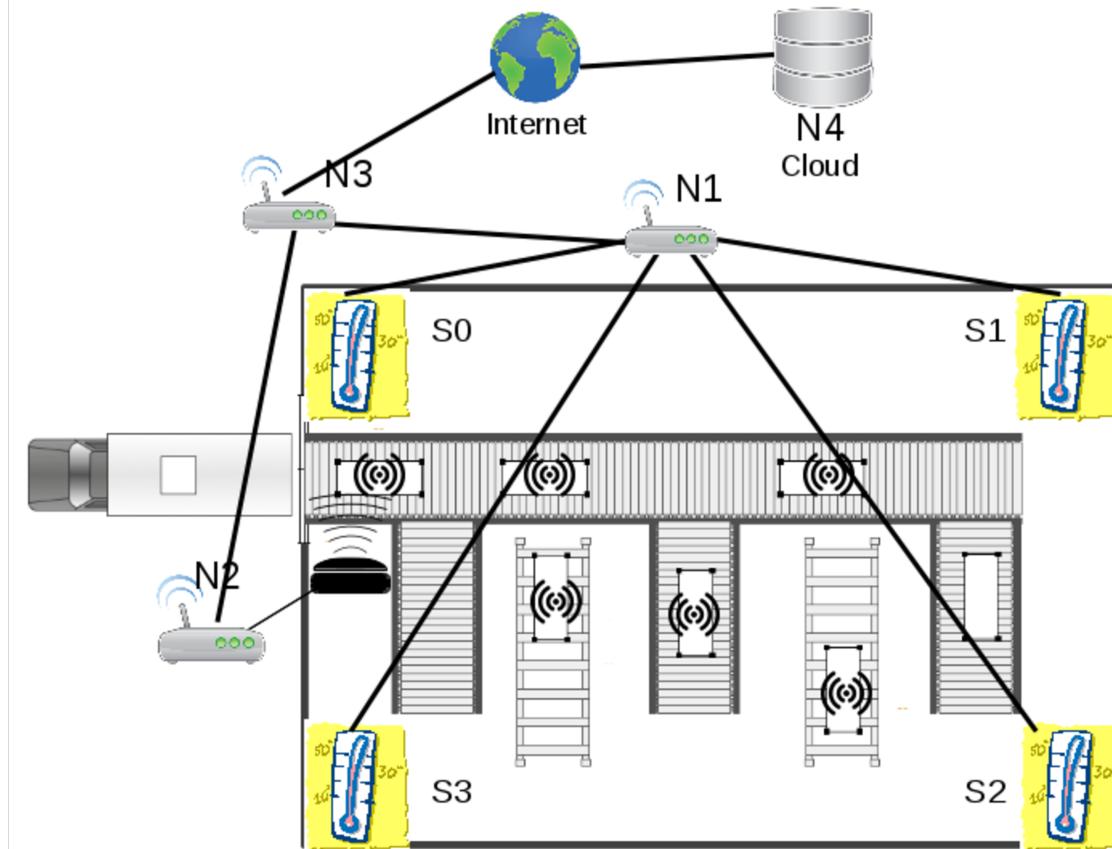
1.

A view of IoT-LySa

Designing a smart storehouse



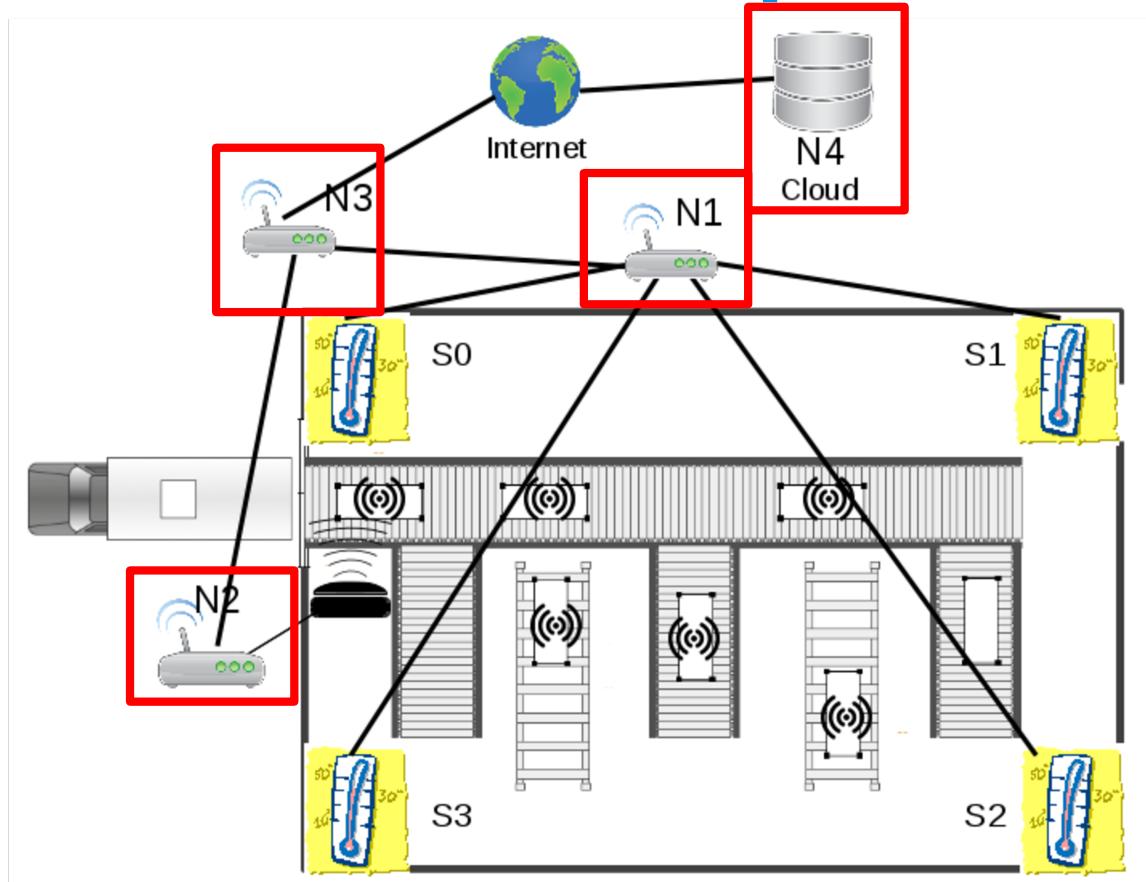
A smart storehouse with perishable food



Supponiamo di avere un magazzino che mi contiene merce deperibile quindi che ha bisogno di stare un tempo limitato e che deve essere soprattutto refrigerata in maniera opportuna. Quindi supponiamo di avere quattro sensori ai quattro angoli del magazzino e controllati da altrettanti nodi e supponiamo di avere un nodo che controlla tutti e quattro i sensori e che quindi sostanzialmente controlla la temperatura dell'edificio perché che cosa fa? Tiene la temperatura sotto controllo eventualmente parlandone con i nodi n2 e n3, abbiamo quindi quattro sensori e poi abbiamo naturalmente degli attuatori che azionano e che comandano sostanzialmente il sistema di raffreddamento.



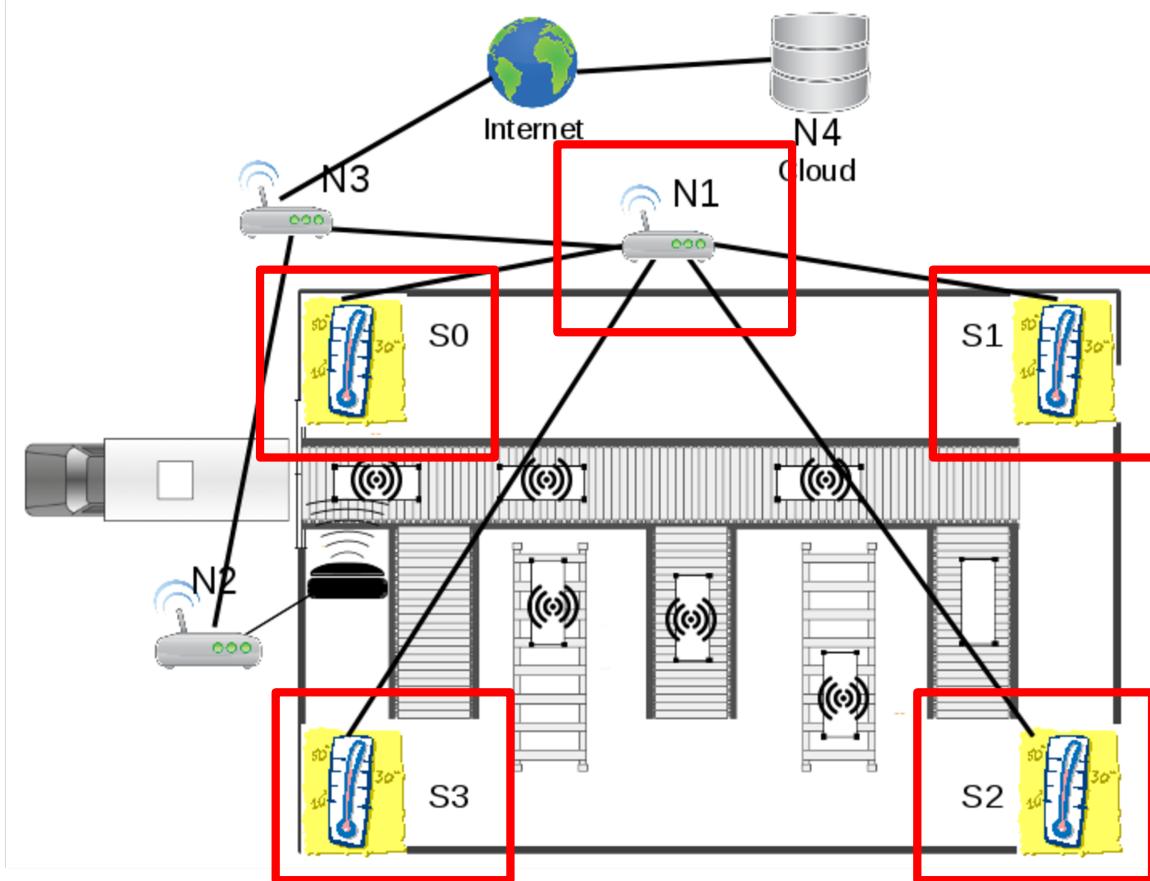
A smart storehouse with perishable food



Four components:
N1, N2, N3, N4



A smart storehouse with perishable food

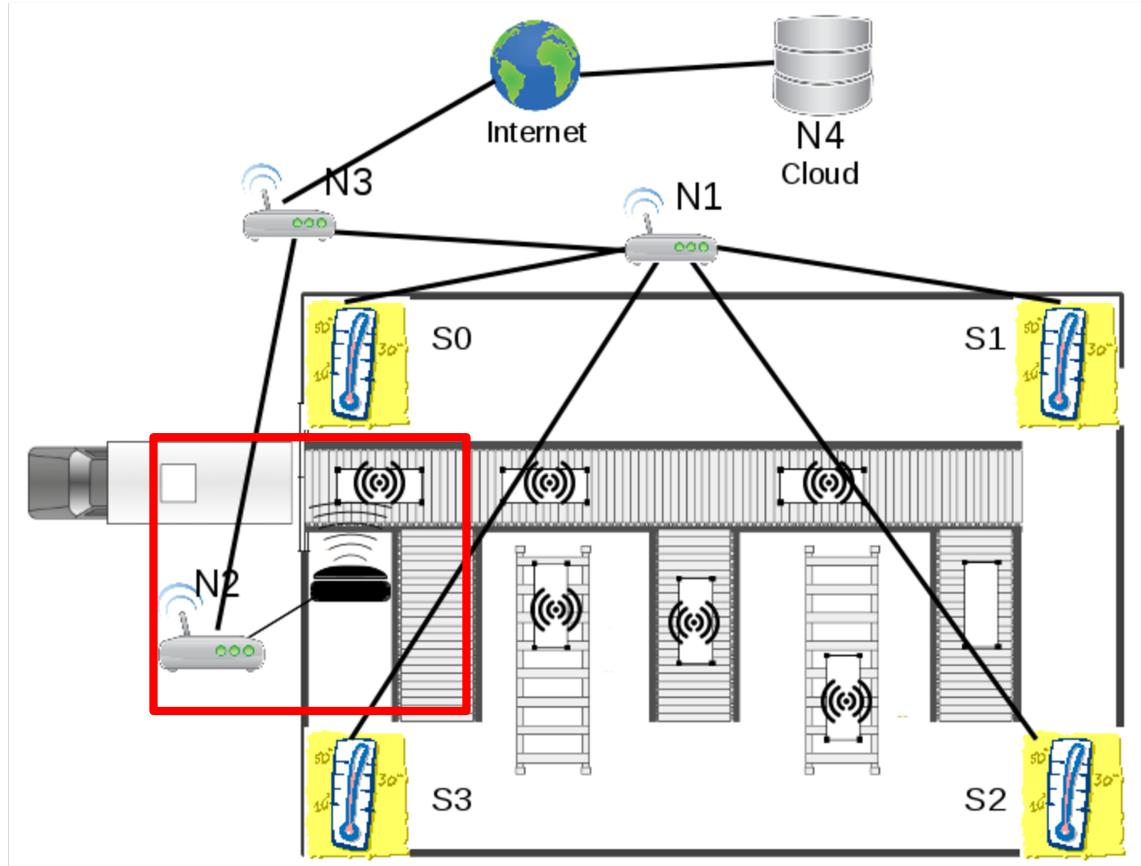


Node N1

- keeps the temperature under control and sends it to N3
- Four sensors S0, S1, S2, S3
- Actuators for controlling the cooling system



A smart storehouse with perishable food

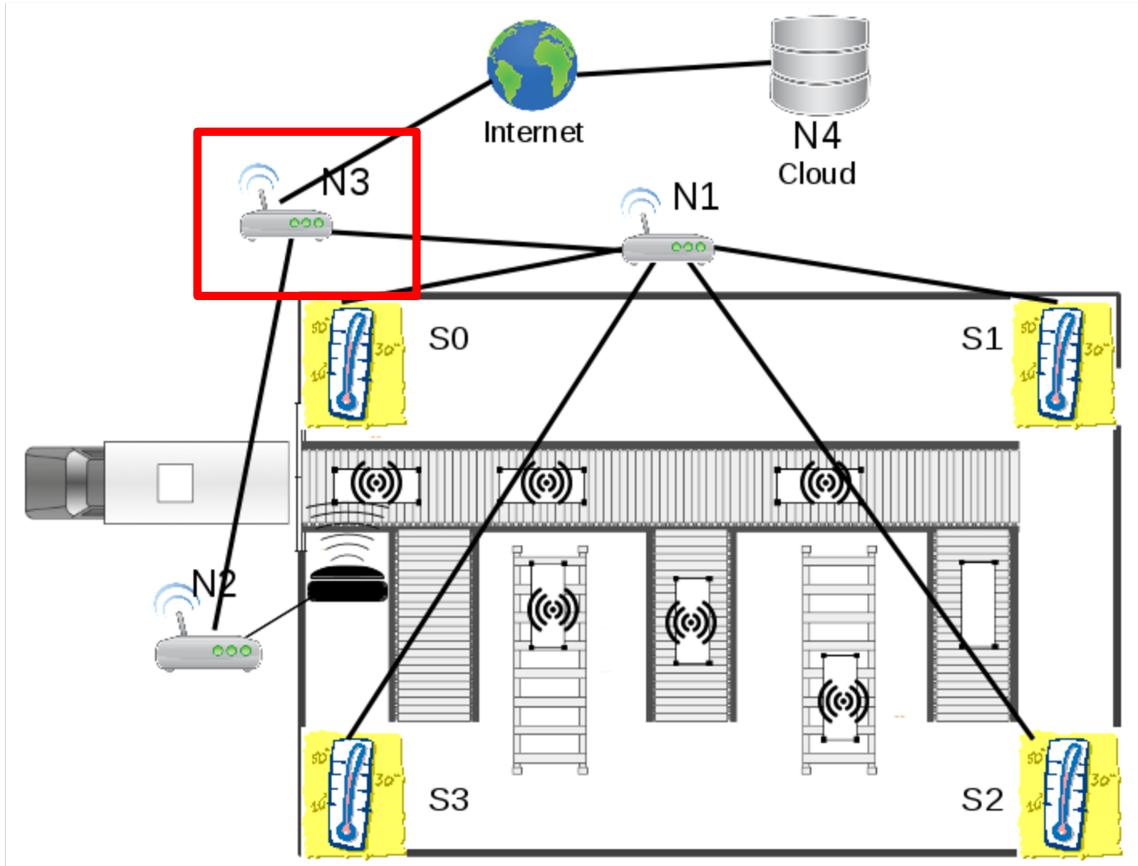


Node N2

- does the stocktaking and sends it to N3
- A RFID reader R0
- Each box with food has a RFID



A smart storehouse with perishable food

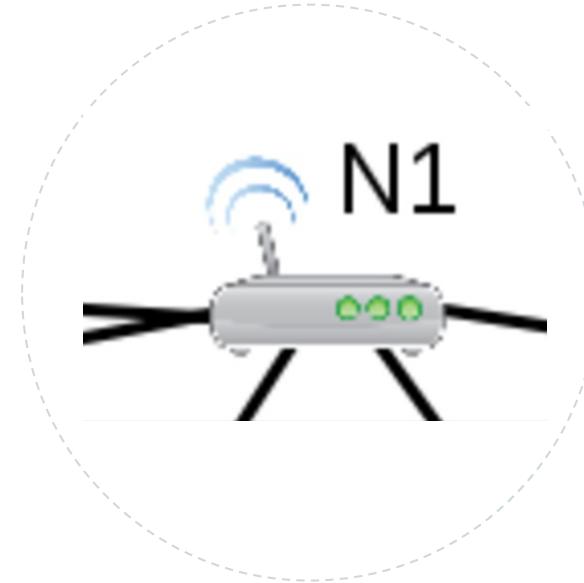


Node N3

- stores the stocktaking in the Cloud N4
- checks if the temperature is acceptable for the quantity and kind of food
- drives N1

Il nodo n3 va a fare lo storing di quanto c'è, cioè va a controllare sostanzialmente quanta merce c'è all'interno del magazzino e va a controllare dopo che gli è stata mandata la media della temperatura all'interno del magazzino se quella temperatura è sufficiente per mantenere intatto il quantitativo di merce che sto toccando all'interno del magazzino.

IoT-LySa specification



$$N_1 = \ell_1 : [\Sigma_1 \parallel P_c \parallel (S_0 \parallel S_1 \parallel S_2 \parallel S_3)]$$

A questo punto la specifica quindi la possiamo vedere come abbiamo visto prima, quindi ogni nodo arriva con la sua bella etichetta all'interno ancora della sua memoria locale e il suo processo che gestisce la logica e tutti gli eventuali sensori quindi in questo caso abbiamo che c'è un solo nodo che raccoglie come sensori s0 s1 s2 s3 quindi l' identificatore è questo qua la shell store è questa, questo è il processo di controllo e questi sono i sensori che gestiscono sostanzialmente la temperatura.

IoT-LySa specification

Node id

$$N_1 = \ell_1 : [\Sigma_1 \parallel P_c \parallel (S_0 \parallel S_1 \parallel S_2 \parallel S_3)]$$



IoT-LySa node specification

Node id

Shared store

$$N_1 = \ell_1 : [\Sigma_1 \parallel P_c \parallel (S_0 \parallel S_1 \parallel S_2 \parallel S_3)]$$



IoT-LySa node specification

Node id

Shared store

$$N_1 = \ell_1 : [\Sigma_1 \parallel P_c \parallel (S_0 \parallel S_1 \parallel S_2 \parallel S_3)]$$

Control process



IoT-LySa node specification

Node id

Shared store

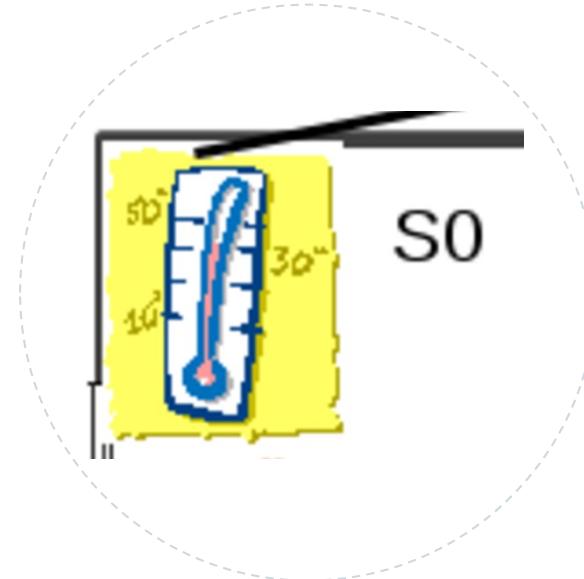
$$N_1 = \ell_1 : [\Sigma_1 \parallel P_c \parallel (S_0 \parallel S_1 \parallel S_2 \parallel S_3)]$$

Control process

Temperature
sensors



Sensor specification



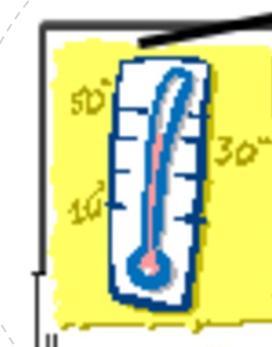
$$S_0 = \mu h. 0 := v. \tau. h$$

Quindi ogni sensore a questo punto non è più sorprendente, semplicemente inserisce nel suo identificatore riservato il valore rilevato in quel momento nel suo angolo di magazzino.

Sensor specification

Iteration

$$S_0 = \mu h. 0 := v. \tau. h$$

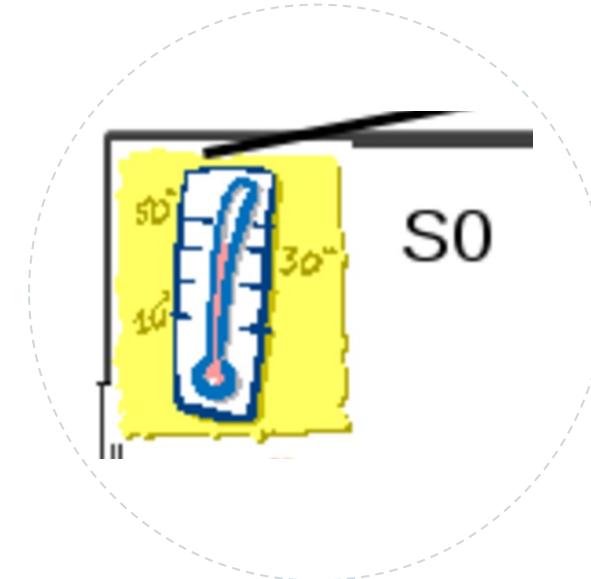


Sensor specification

Internal activities

Iteration

$$S_0 = \mu h. 0 := v. \tau. h$$



La comunicazione con il processo avviene attraverso la shared store, quindi una volta che il sensore ha rilevato il suo valore, poi all'interno n1 andrà a copiarsi il valore dentro la memoria locale

Sensor specification

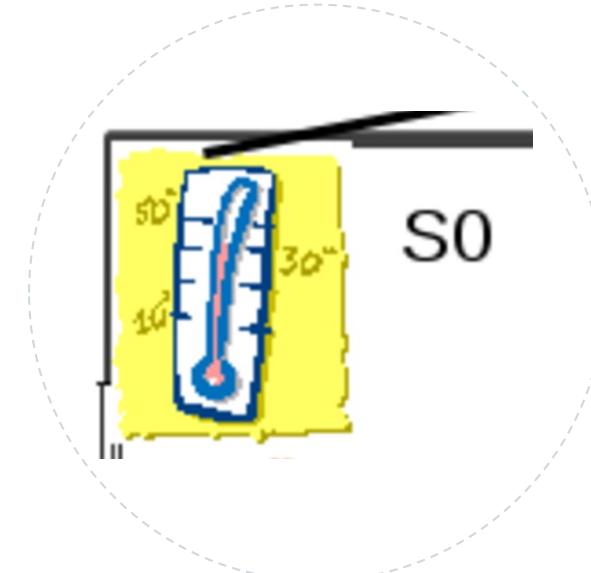
Internal activities

Iteration

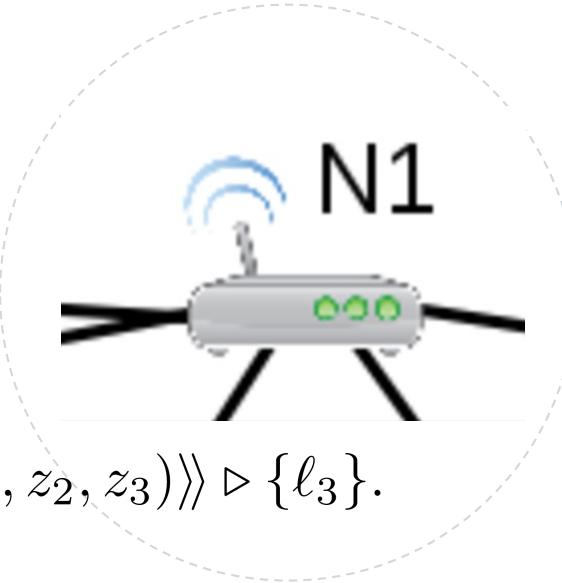
$$S_0 = \mu h. 0 := v. \tau. h$$

Sensing a value

- Communication with processes through a shared store
- Each sensor has a reserved location (also sensor id)



Control process specification



$$P_c = \mu h.z_0 := 0.z_1 := 1.z_2 := 2.z_3 := 3.\langle\langle \text{avg}(z_0, z_1, z_2, z_3) \rangle\rangle \triangleright \{\ell_3\}.$$

$$th_1 - th_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 + th_3 ?$$

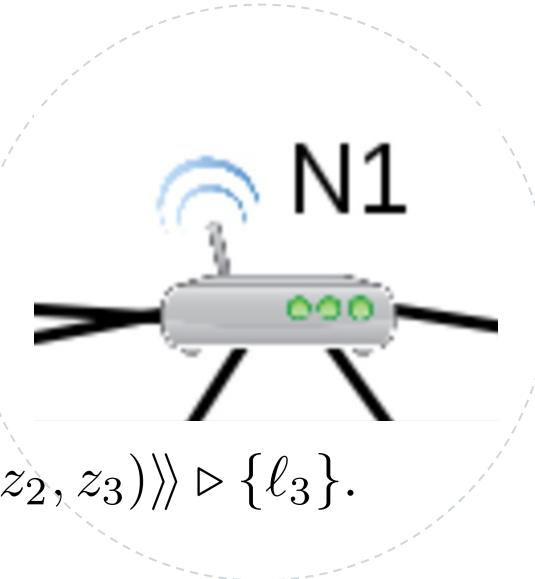
$$th_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 ? h$$

: $\langle j, start \rangle . h$

: $\langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}.h$

infatti questo è il processo che controlla tutti i sensori che cosa fa? Si va a copiare in z_0 z_1 z_2 z_3 i valori raccolti dai sensori, quindi zero semplicemente sta copiando i valori associati all'identificatore riservato del sensore zero lo va a copiare in z_0 . Una volta che ha tutti e quattro i valori dei sensori, semplicemente fa quello che uno si aspetta, cioè va a calcolare la media, va a calcolare la media che spedisce poi più avanti, ma sostanzialmente avendo la media è in grado già lui di fare le sue valutazioni, quindi in particolare va a controllare se la media delle temperature sta nel range che serve al magazzino e che cosa succede. Se sta nel range ed è tutto tranquillo, va a controllare se è il caso di fare delle azioni di controllo sul sistema di refrigerazione di condizionamento, quindi, in particolare se deve abbassare o alzare l'impianto, in modo da garantire la temperatura ideale, altrimenti se siamo sopra la soglia in qualche modo si rileva una anomalia grossa e allora si deve avvertire il processo successivo in modo che intervenga per sistemare la situazione.

Control process specification



$$P_c = \mu h.z_0 := 0.z_1 := 1.z_2 := 2.z_3 := 3.\langle\langle \text{avg}(z_0, z_1, z_2, z_3) \rangle\rangle \triangleright \{\ell_3\}.$$

$$th_1 - th_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 + th_3 ?$$

$$th_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 ? h$$

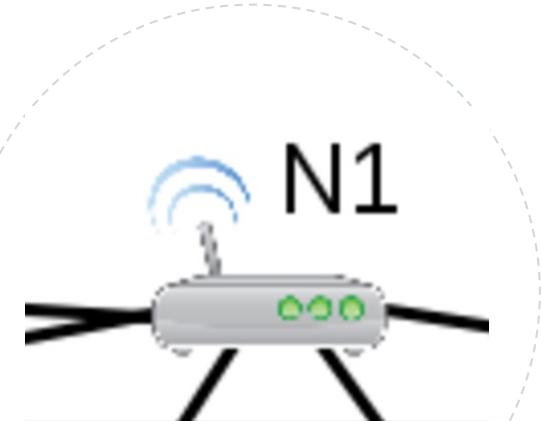
: $\langle j, \text{start} \rangle . h$

: $\langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}. h$

Store current sensor values into local variables



Control process specification



$$P_c = \mu h.z_0 := 0.z_1 := 1.z_2 := 2.z_3 := 3.\langle\langle \text{avg}(z_0, z_1, z_2, z_3) \rangle\rangle \triangleright \{\ell_3\}.$$

$$th_1 - th_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 + th_3 ?$$

$$th_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 ? h$$

: $\langle j, \text{start} \rangle . h$

: $\langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}. h$

Compute the average and send it to node N3



Control process specification

$$P_c = \mu h.z_0 := 0.z_1 := 1.z_2 := 2.z_3 := 3.\langle\langle \text{avg}(z_0, z_1, z_2, z_3) \rangle\rangle \triangleright \{\ell_3\}.$$

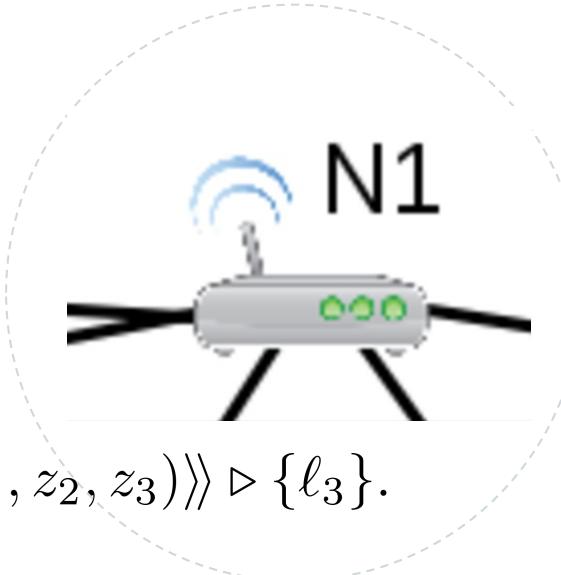
$$\text{th}_1 - \text{th}_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq \text{th}_2 + \text{th}_3 ?$$

$$\text{th}_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq \text{th}_2 ? h$$

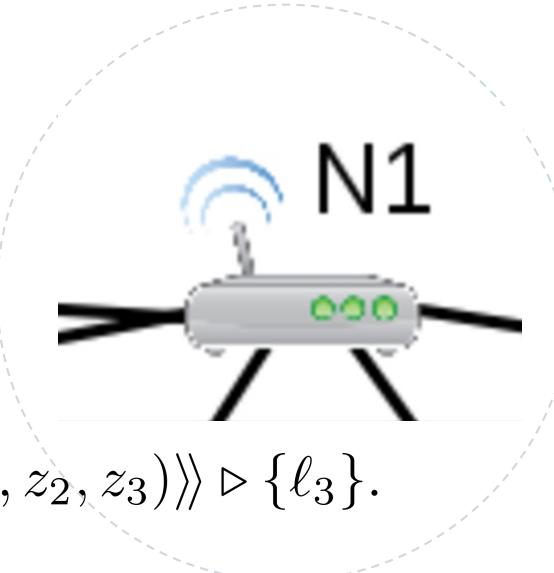
: $\langle j, \text{start} \rangle . h$

: $\langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}. h$

If the temperature is out of the admissible range $[\text{th}_1, \text{th}_2]$ of a value greater than th_3 , raise the alarm



Control process specification



$$P_c = \mu h.z_0 := 0.z_1 := 1.z_2 := 2.z_3 := 3.\langle\langle \text{avg}(z_0, z_1, z_2, z_3) \rangle\rangle \triangleright \{\ell_3\}.$$

$$th_1 - th_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 + th_3 ?$$

$$\text{th}_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq \text{th}_2 ? h$$

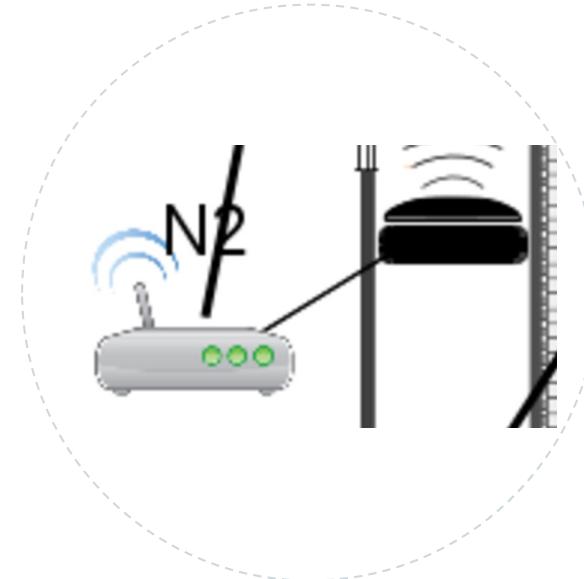
: $\langle j, \text{start} \rangle . h$

$$: \langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}. h$$

If the temperature is out of the admissible range $[th_1, th_2]$ of a value less than th_3 , then start the cooling system

Naturalmente la media viene mandata anche al nodo N3 perché abbiamo visto che è quello che deve controllare se la temperatura che c'è in quel momento in magazzino è quella giusta per conservare quel quantitativo di merce che lui ha controllato esserci in quel momento dentro e quindi c'è questo passaggio ulteriore. Quindi se la temperatura è fuori dal range ammissibile, allora si deve far scattare l'allarme, altrimenti se è all'interno del range ammissibile si deve far partire il condizionamento.

Other nodes: stocktaking



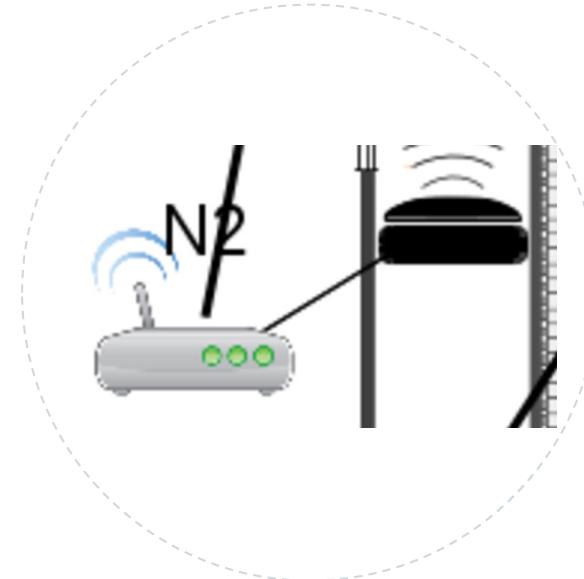
$$N_2 = \ell_2 : [\Sigma_2 \parallel R_0 \parallel \mu h. x := 0. db := update(db, x). \tau. h \parallel \\ \mu h. \langle\langle db \rangle\rangle \triangleright \{\ell_3\}. h]$$



Other nodes: stocktaking

Shared store +
RFID Reader
(as before)

$$N_2 = \ell_2 \cdot [\Sigma_2 \parallel R_0 \parallel \mu h. x := 0. db := update(db, x). \tau.h \parallel \\ \mu h. \langle\langle db \rangle\rangle \triangleright \{\ell_3\}. h]$$

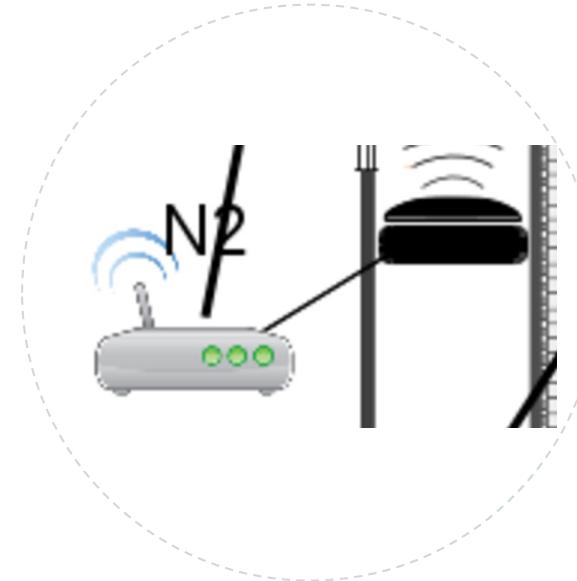


Other nodes: stocktaking

Shared store +
RFID Reader
(as before)

Update the
stocktaking with
incoming food

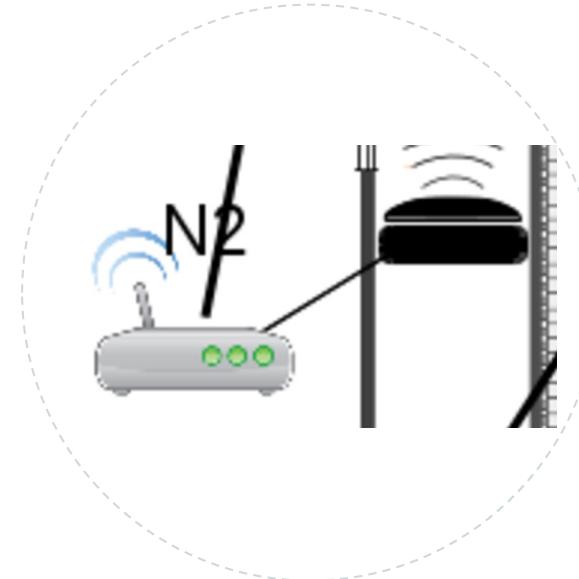
$$N_2 = \ell_2 : [\Sigma_2 \parallel R_0 \parallel \mu h. x := 0. db := update(db, x). \tau. h \parallel \\ \mu h. \langle\langle db \rangle\rangle \triangleright \{\ell_3\}. h]$$



Other nodes: stocktaking

Shared store +
RFID Reader
(as before)

Update the
stocktaking with
incoming food



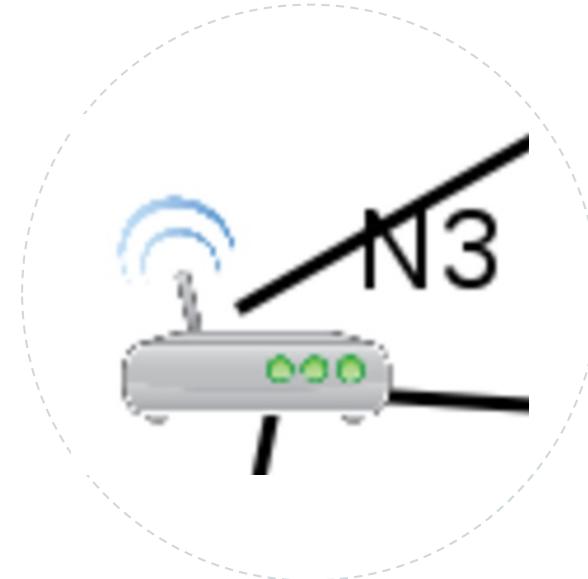
$$N_2 = \ell_2 : [\Sigma_2 \parallel R_0 \parallel \mu h. x := 0. db := update(db, x). \tau. h \parallel$$

$$\mu h. \langle\langle db \rangle\rangle \triangleright \{\ell_3\}. h]$$

Send the stocktaking
to N3

Il nodo N2 va a controllare invece attraverso gli RFID (si suppone che ogni elemento che arriva nel magazzino sia corredato da rfid) per cui questo nodo va a controllare sostanzialmente cosa entra nel magazzino e fare l' update di supponiamo che ci sia una sorta di database e quindi manda il risultato del database al nodo N3.

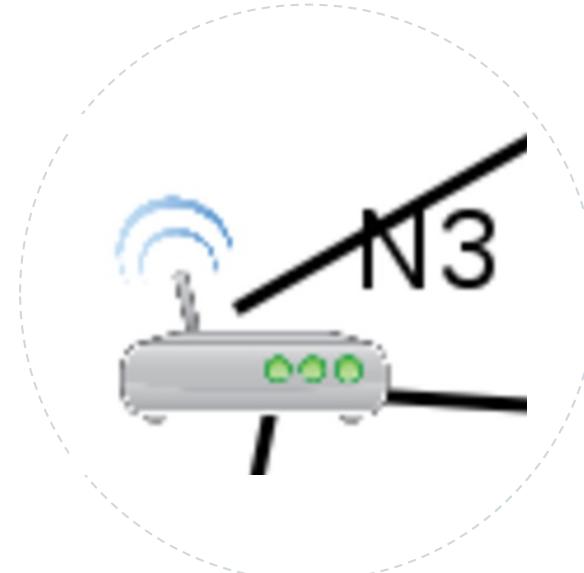
Other nodes: checking temperature



$\mu h.\text{temp} \notin \text{validRange}(db) ? \langle\langle \text{outRange} \rangle\rangle \triangleright \{\ell_4\}.$
 $\langle\langle \text{validRange}(db) \rangle\rangle \triangleright \{\ell_1\}.h$
: h



Other nodes: checking temperature



$\mu h.\text{temp} \notin \text{validRange}(db) ? \langle\langle \text{outRange} \rangle\rangle \triangleright \{\ell_4\}.$
 $\langle\langle \text{validRange}(db) \rangle\rangle \triangleright \{\ell_1\}.h$

: h

If the temperature is not appropriate for the stored food,
then record this event in the Cloud and instruct N1

Il nodo N3 tra le altre cose che cosa fa? Va a vedere se la temperatura che ha ottenuto sta nel range valido per quel tipo di merce, quindi se la temperatura non è appropriata per il cibo che è stato stoccati all'interno del magazzino, questo evento viene mandato nel cloud e si deve in qualche maniera gestire la situazione di anomalia e comunque deve essere gestito per sistemarlo, altrimenti

Cosa dobbiamo controllare, per esempio in questo caso allora l' intuizione è che io di nuovo cerchi di capire qual è il passaggio dei dati astratti, quindi in qualche maniera li segua in maniera simbolica e capisca dov'è che ci possono essere vulnerabilità di sicurezza? Ora l'analisi che cosa fa? L'analisi calcola ,di nuovo le componenti sono quelli che abbiamo già visto, la abstract store (sigma cappuccio) il network environment k e la collezione di dati theta maiuscolo. Ora qual è il l'aggiunta? Stabilire come tracciare questi dati astratti. Allora ritorniamo un attimo, all' esempio della slide 13, io voglio da una parte essere sicura che i dati su cui io faccio le mie considerazioni che mi portano a azionare l'impianto di condizionamento o a generare una situazione d'allarme che va sistemato e tutto, quello che mi serve è un'informazione importante, voglio sapere quanto sono affidabili i dati che vengono dai sensori, cioè può essere che questi sensori possono essere in qualche maniera manomessi, tampered e quindi il tipo di valutazione che faccio è inficiata dal fatto che questi sensori mi danno dei valori che non sono quelli corretti? L'altra parte dell'analisi è quella che mi dice e che in questo caso però qui non la vediamo, ci sono delle parti di dati che vorrei tenere protetti? Quindi come faccio a vedere se questi dati vengono protetti dal mio sistema, cioè quand'è che passano in chiaro oppure no? L'analisi che stiamo cercando di mettere su in questo momento è proprio un'analisi che mi traccia, tipo taint analysis, questo tipo di informazioni quindi da una parte i dati che possono essere tampered e dall'altra i dati che invece sono sensibili e vanno protetti. Come cambiare sostanzialmente i valori astratti e allo stesso tempo le regole dell'analisi che mi fanno propagare questi valori astratti all'interno della rete, in modo da tracciare informazioni di questo tipo, cioè in particolare cominciamo con uno, supponiamo di voler tracciare i sensori cioè le informazioni che vengono da sensori che possono essere danneggiati tempered? Potremmo sfruttare un'idea simile a quella dei dati segreti, questo sicuramente per quanto riguarda la parte della secrecy, e quindi dei dati che vanno protetti, quindi io potrei fare l'analisi che abbiamo fatto fino a ora e segnare come segreti e pubblici i vari dati astratti e poi vedere come fluendo all'interno di composizioni, quindi di aggregazioni e cose varie questi dati rimangono protetti o meno e quindi questa è la prima idea, quindi come faccio a propagare la segretezza? Come abbiamo visto prima, io stabilisco che ne so che alcuni dati dei sensori sono segreti e quindi come decido come si propaga questa informazione rispetto a ipossibili modi in cui i dati si propagano, in quali forme si propagano, in quanto espressioni? Potendo sfruttare il concetto che se abbiamo delle informazioni pubbliche ma andiamo inserire anche una sola informazione privata tutto dobbiamo trattarlo come segreto, esatto, quindi il drop of strict rende tutto secret. In questa maniera l'unico modo per avere un sistema segreto è non comunicare mai segreti invece poi serve comunicare segreti quindi com'e che devo aggiustarlo? Con l' encryption, quindi io posso dire che una tupla con almeno un elemento segreto diventa segreta tranne nel caso in cui questa tupla viene protetta da un encryption, a quel punto l'elemento è segreto, ma la tupla è criptata e cifrata e diventa di nuovo pubblica perché è come se si opacizza l'interno e quindi tutto ciò che è all'interno non mi importa più se è segreto o pubblico. Invece il tampering segue una via diversa, quindi io anche in questo caso propago l'informazione, quindi se parto con l'idea che qualcosa può essere tamper able, questo tamperable mi arriva anche lì, nei livelli più alti, quindi se c'ho un qualcosa che è tamperable una tupla che contiene una cosa tamperable vuol dire che in qualche modo la funzione che io applico a questi dati qualche problema potrebbe averlo. Quindi anche qui posso ipotizzare che magari potremmo marcare dei sensori che possono essere tampered e trattare tutti i nodi che metto in conseguenza come tali, bisogna esattamente anche in questo caso taggare le cose che possono essere a rischio e poi propagare l'informazione, quindi stabilendo anche di una policy per cui una cosa che è tamper potrebbe essere sanitizzata o non so, in quel caso lì di nuovo risistemo qualcosa che invece può essere rischiosa, quindi uno potrebbe decidere che il tamperable galleggia sempre più in alto, però poi se so che qualcosa è tamperable potrei decidere di fare qualche azione per renderlo invece affidabile, a cui potrei controllare se i valori sono quelli giusti o cose di questo genere. Quindi quello che facciamo è associare l'informazione, come diceva il vostro collega, ai singoli valori astratti e far fluire quelli in modo da decidere di che cosa ho bisogno e come sta andando la mia comunicazione. In particolare, quindi io è come se ai valori astratti aggiungessi una componente, un'etichetta che mi dice se il valore astratto ha componenti di tipo tamperable oppure componenti da proteggere, quindi sensitive che sono due tipi di informazioni diverse, poi vediamo chi li possiamo in qualche maniera associare e avendo valori astratti che hanno questo specie di bollino accanto io posso decidere in quale modo si propagano, con quale regole e quindi , l'analisi rimane la stessa, solo che si porta dietro anche questa ulteriore informazione che è quella che mi consente di stabilire in ogni punto i dati, cioè in ogni nodo quali sono i valori calcolati, questi valori calcolati posso andare a vedere di che natura siano, cioè che componenti hanno e che componenti hanno in particolare di tipo possibly tamperable o possibly sensitive. Una volta che ho i dati etichettati con il loro livello di sicurezza da questo punto di vista è chiaro che posso fare il passo ulteriore, andare a vedere se ci sono dei punti dove ho un'istruzione condizionale dalla quale dipende poi un'attuazione e che si basa su questi dati. Quindi se io devo accendere il sistema di condizionamento del mio magazzino perché è arrivata ancora merce e quindi il livello di refrigerazione non è garantito e subisco un attacco da un corrente che vuol far sì che la mia roba vada male quello che succede è che devo controllare se c'è qualcosa che può essere alterato perché questo mi mette al riparo da possibili conseguenze nefaste, quindi se scopro che uno dei sensori in un punto è accessibile a chiunque, io devo fare in modo di capire che questo mi implica delle conseguenze gravi sulla mia organizzazione e se scopro che c'è questo problema, allora posso prendere le mie dovute precauzioni che, per esempio, nel caso del magazzino, potrebbero essere: qualsiasi media io abbia possa andare a vedere se nella stessa zona del magazzino i valori rilevati dai due sensori sono paragonabili quindi se ce n'è uno fuori scala mi devo far venire il dubbio che qualcosa può essere andato storto.

Abstract values

Trees with a finite depth d

$$\hat{\mathcal{V}} \ni \hat{v} ::= \nu^b$$

$$\{\hat{v}_1, \dots, \hat{v}_n\}_{k_0}^b$$

$$\top^b$$

Quindi ho i soliti valori astratti che come dicevo vengono corredati da una seconda componente che possiamo scrivere come b e che è una sorta di etichetta che mi dice che razza di natura abbia questo dato, quindi in particolare b può avere tre forme: abbiamo un dato che con questo quadratino rosso mi dice, guarda, attenzione, questo è un dato tanto tamperable, non vuol dire che è per forza manomesso, ma che potrebbe essere manomesso. Il quadratino azzurro mi dice che il dato è sensibile, quindi in qualche modo va protetto. Poi abbiamo l'unione delle due cose che viene indicata dal quadratino con la croce che sta per tamperable ma anche sensitive e poi invece abbiamo il quadratino vuoto nero che sta per untainted quindi su questo dato non ho nessuna indicazione né rispetto alla sensitive né rispetto alla temperability.

clear data

encrypted data

cut

And with a tag b

◊untainted

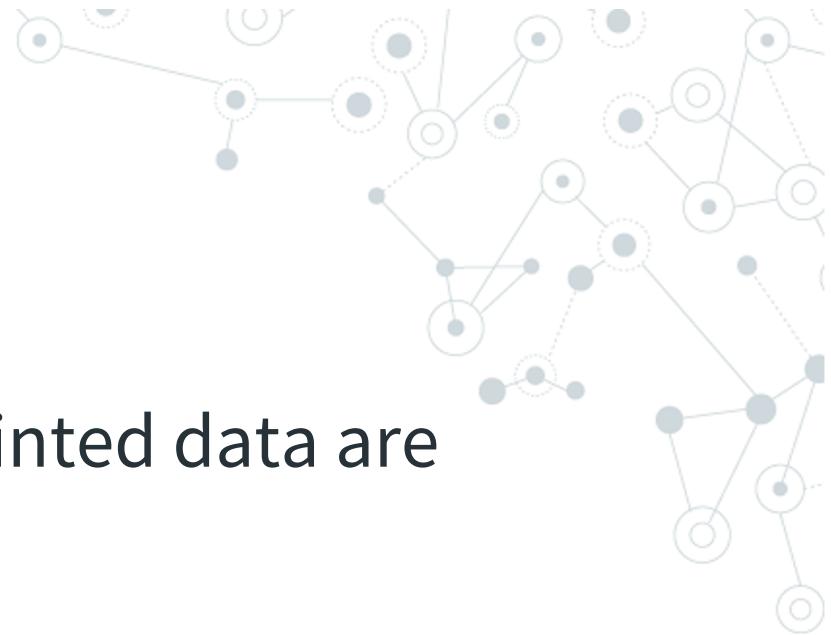
◆ tamperable

◆ sensitive

◆ tamperable + sensitive



Default taint propagation policy



- Values computed from tainted data are tainted
- Encryption declassifies sensitive data
- Designers can provide their policies for some functions

A questo punto quello che faccio è semplicemente calcolare la mia analisi portandomi dietro questa componente, quindi, la cosa che devo fare è sostanzialmente stabilire la politica di propagazione perché naturalmente io posso etichettare tutti questi elementi, però voglio capire una volta che sto aggregando un valore con un'etichetta ad un'altro che ne ha un'altra, cosa succede? Quindi come combino valori di questo tipo in maniera tale da propagare come ci si aspetta l'informazione sia sulla temperability che sulla sensitive. A questo punto quindi penso quali sono le politiche di combinazione e la prima politica che è già venuta in mente e già discussa prima è quella del fatto che naturalmente, anche se ho dei dati sensibili, l'encryption me li declassifica a valori che possono passare in chiaro perché naturalmente ci pensa l'encryption a proteggerli. Così come a occhio se ho dei valori che sono composti da valore tainted più o meno dovrei mantenere il fatto che sono tali.



Using static taint analysis: ingredients

1. Classify tainted data sources
2. Determine critical points
3. Define custom taint propagation policies

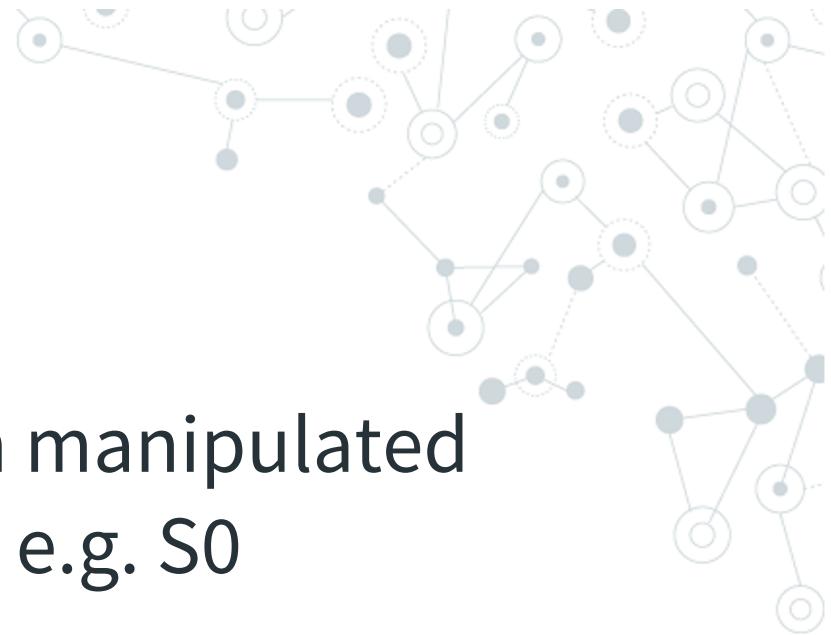
Quindi quello che succede è che la mia analisi mi aiuta a classificare le sorgenti di tipo tainted, esattamente come diceva giuseppe, quindi io posso marcare o no i sensori, naturalmente adesso io ho parlato di sensori ma nella stessa maniera, uno può pensare a nodi che non sono affidabili, che quindi potrebbero ricevere delle informazioni buona dal senso di affidabili ma ad arte poi la possono manipolare, quindi chiaramente è qualcosa che si può gestire anche in questo caso, e poi che devo fare? Devo andare a controllare quali sono i punti possibilmente critici in cui prendo decisioni per vedere appunto qual è l'impatto di valori non esattamente affidabili per prendere decisioni e quindi naturalmente mi devo anche, come dicevamo prima, inventare quali sono le politiche di propagazione di questi colori e mano a mano che i dati fluiscono attraverso il nodo, magari in forma composita, con aggregazione e cose del genere.

Static taint analysis in our example

1. Assume an attacker can manipulate sensor only one sensor, e.g. S0

Its values are tagged with 

2. The critical point is the test in N3
 $(temp \notin validRange(db))^a$
3. Default taint propagation policy



Si può pensare anche a un nodo che non è trusted per semplicità, supponiamo che, per esempio, un attaccante possa manipolare un sensore e quindi s0, questo pensiamo nell'esempio naturalmente, s0 ha quindi tutti i valori che produce le produce con etichetta rossa che sta per tamperable. Allora io vado a vedere se esiste un punto della mia computazione, dove la decisione di attuazione dipende da questo valore e quindi se questo succede vuol dire che può esserci un problema, quindi devo andare a controllare se decisioni critiche dipendono da fonti che non sono affidabili.

Analysis of N1



$$P_c = \underline{\mu h}. z_0 := 0. z_1 := 1. z_2 := 2. z_3 := 3. \langle\langle \text{avg}(z_0, z_1, z_2, z_3) \rangle\rangle \triangleright \{\ell_3\}.$$

$$th_1 - th_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 + th_3 ?$$

$$th_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 ? h$$

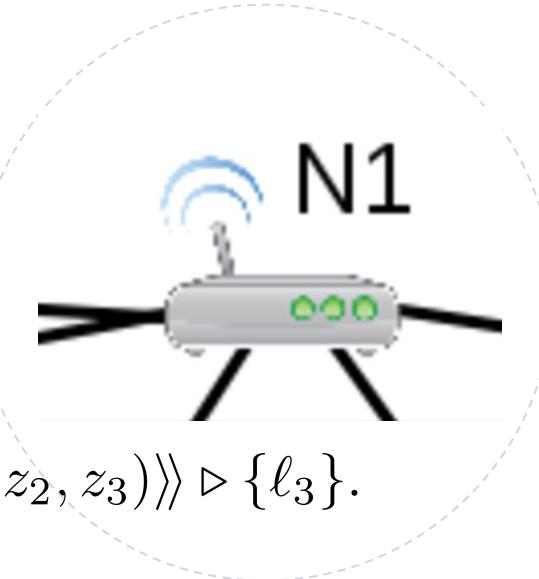
: $\langle j, \text{start} \rangle. h$

: $\langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}. h$

	0	1	2	3	z_0	z_1	z_2	z_3
$\hat{\Sigma}$	◆	◆	◆	◆				



Analysis of N1



$$P_c = \mu h.z_0 := 0.z_1 := 1.z_2 := 2.z_3 := 3.\langle\langle \text{avg}(z_0, z_1, z_2, z_3) \rangle\rangle \triangleright \{\ell_3\}.$$

$$th_1 - th_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 + th_3 ?$$

$$th_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 ? h$$

: $\langle j, \text{start} \rangle . h$

: $\langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}.h$

	0	1	2	3	z_0	z_1	z_2	z_3
$\hat{\Sigma}$	◇	◇	◇	◇	◇			

Per esempio nel nostro caso, supponiamo di avere appunto l'analisi del processo che gestisce quattro sensori di temperatura che calcola la media, supponiamo che il sensore zero sia inaffidabile, quindi tamperabile e quindi mentre tutti gli altri li classifico con il quadratino nero che vuol dire niente da dichiarare, quello lì lo metto da parte e lo taggo e dico attenzione a quello potrebbe non essere affidabile.

Analysis of N1



$$P_c = \mu h.z_0 := 0.z_1 := 1.z_2 := 2.z_3 := 3.\langle\langle \text{avg}(z_0, z_1, z_2, z_3) \rangle\rangle \triangleright \{\ell_3\}.$$

$$th_1 - th_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 + th_3 ?$$

$$th_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 ? h$$

: $\langle j, \text{start} \rangle . h$

: $\langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}.h$

	0	1	2	3	z_0	z_1	z_2	z_3
$\hat{\Sigma}$	◇	◇	◇	◇	◇	◇		



Analysis of N1



$$P_c = \mu h.z_0 := 0.z_1 := 1.z_2 := 2.z_3 := 3.\langle\langle \text{avg}(z_0, z_1, z_2, z_3) \rangle\rangle \triangleright \{\ell_3\}.$$

$$th_1 - th_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 + th_3 ?$$

$$th_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 ? h$$

: $\langle j, \text{start} \rangle . h$

: $\langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}.h$

	0	1	2	3	z_0	z_1	z_2	z_3
$\hat{\Sigma}$	◇	◇	◇	◇	◇	◇	◇	



Analysis of N1



$$P_c = \mu h.z_0 := 0.z_1 := 1.z_2 := 2.z_3 := 3.\langle\langle \text{avg}(z_0, z_1, z_2, z_3) \rangle\rangle \triangleright \{\ell_3\}.$$

$$th_1 - th_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 + th_3 ?$$

$$th_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 ? h$$

: $\langle j, \text{start} \rangle . h$

: $\langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}.h$

	0	1	2	3	z_0	z_1	z_2	z_3
$\hat{\Sigma}$	◇	◇	◇	◇	◇	◇	◇	◇



Analysis of N1



$$P_c = \mu h.z_0 := 0.z_1 := 1.z_2 := 2.z_3 := 3.\langle\langle \text{avg}(z_0, z_1, z_2, z_3) \rangle\rangle \triangleright \{\ell_3\}.$$

$$th_1 - th_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 + th_3 ?$$

$$th_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 ? h$$

: $\langle j, start \rangle . h$

: $\langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}.h$

	0	1	2	3	z_0	z_1	z_2	z_3
$\hat{\Sigma}$	◇	◇	◇	◇	◇	◇	◇	◇

$$\text{avg}(z_0, z_1, z_2, z_3) = \diamond$$



Analysis of N1



$$P_c = \mu h.z_0 := 0.z_1 := 1.z_2 := 2.z_3 := 3.\langle\langle \text{avg}(z_0, z_1, z_2, z_3) \rangle\rangle \triangleright \{\ell_3\}.$$

$$th_1 - th_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 + th_3 ?$$

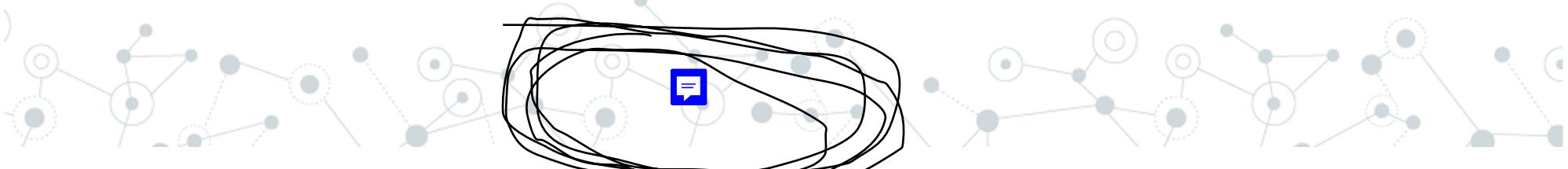
$$th_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 ? h$$

: $\langle j, \text{start} \rangle . h$

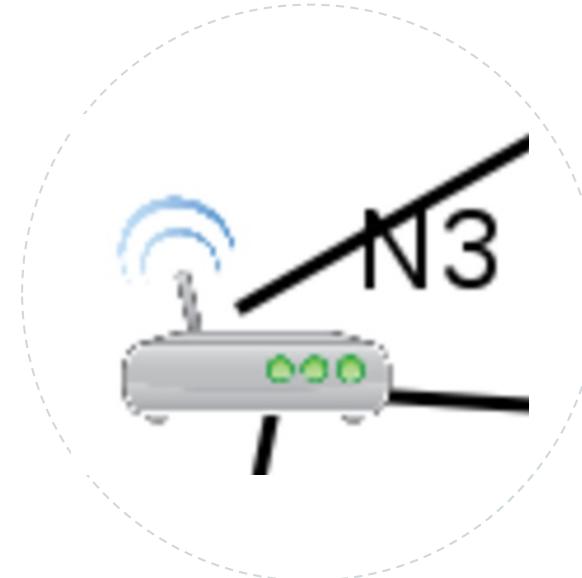
: $\langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}.h$

	0	1	2	3	z_0	z_1	z_2	z_3
$\hat{\Sigma}$	◇	◇	◇	◇	◇	◇	◇	◇

$$\text{avg}(z_0, z_1, z_2, z_3) = \diamond \quad (\ell_1, \diamond) \in \kappa(\ell_3)$$



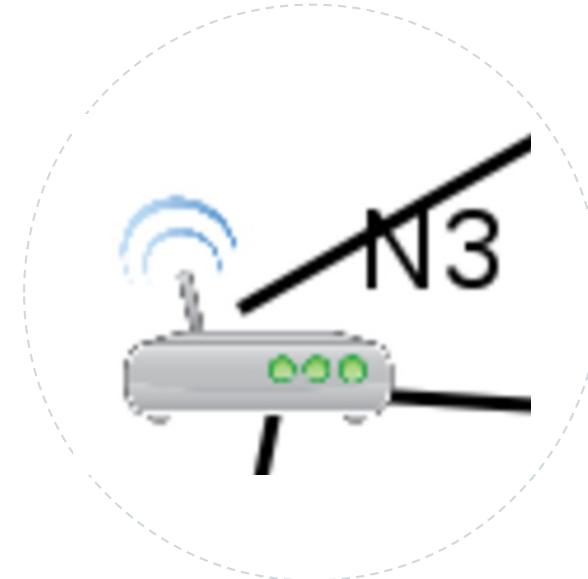
Analysis of N3


$$\mu h. (temp \notin validRange(db))^a ? \langle\!\langle \text{outRange} \rangle\!\rangle \triangleright \{\ell_4\}. \\ \langle\!\langle validRange(db) \rangle\!\rangle \triangleright \{\ell_1\}. h$$

: h

Quindi io mi porto dietro ogni volta l'informazione sul fatto che stanno fluendo cose da un lato all'altro e che alla fine mi ritrovo a utilizzare quest'informazione che ha una parte buggata una parte tainted in un momento in cui devo prendere decisioni se sono dentro al range accettabile oppure no. Quindi ci accorgiamo che c'è almeno un punto nel mio sistema in cui una decisione critica dipende da qualcosa che è tainted e quindi non è affidabile.

Analysis of N3



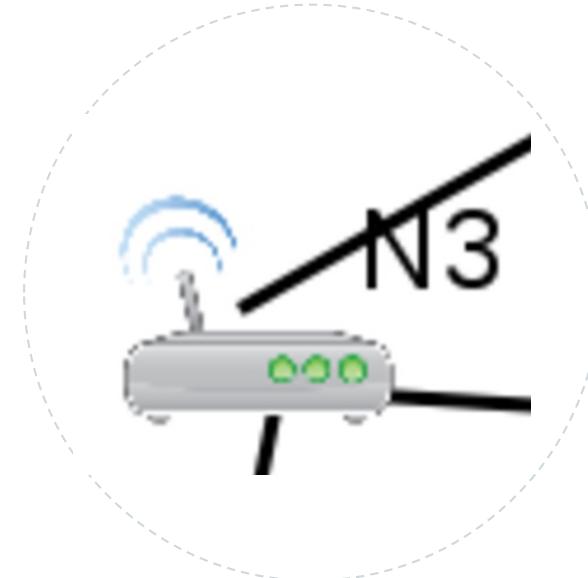
$\mu h.(temp \notin validRange(db))^a ? \langle\langle \text{outRange} \rangle\rangle \triangleright \{\ell_4\}.$
 $\langle\langle validRange(db) \rangle\rangle \triangleright \{\ell_1\}.h$

$(\ell_1, \lozenge) \in \kappa(\ell_3)$

$\hat{\Sigma}(temp) = \lozenge$



Analysis of N3



$\mu h.(temp \notin validRange(db))^a ? \langle\langle \text{outRange} \rangle\rangle \triangleright \{\ell_4\}.$

$\langle\langle validRange(db) \rangle\rangle \triangleright \{\ell_1\}.h$

: h

$(\ell_1, \lozenge) \in \kappa(\ell_3)$

$\hat{\Sigma}(temp) = \lozenge$



\lozenge is used to take decision



An alternative design

Observation 1: Sensors on the same side of the room should perceive the same temperature (with an error of ε)



Observation 2: Consecutive samples of the same sensor should differ of a value δ

We can detect manipulated data and discard them

$$z_0 = \text{adjust}(0, 3, s_0) = \diamond$$



current value

adjacent sensor

previous sample



3.

Conclusion

Summing up

IoT-LySa to specify IoT systems

- ◎ Network of nodes
- ◎ Sensors & actuators
- ◎ Group communication



Tracking data analysis

- ◎ Interaction among nodes
- ◎ Data dependencies and manipulations
- ◎ Taint analysis

