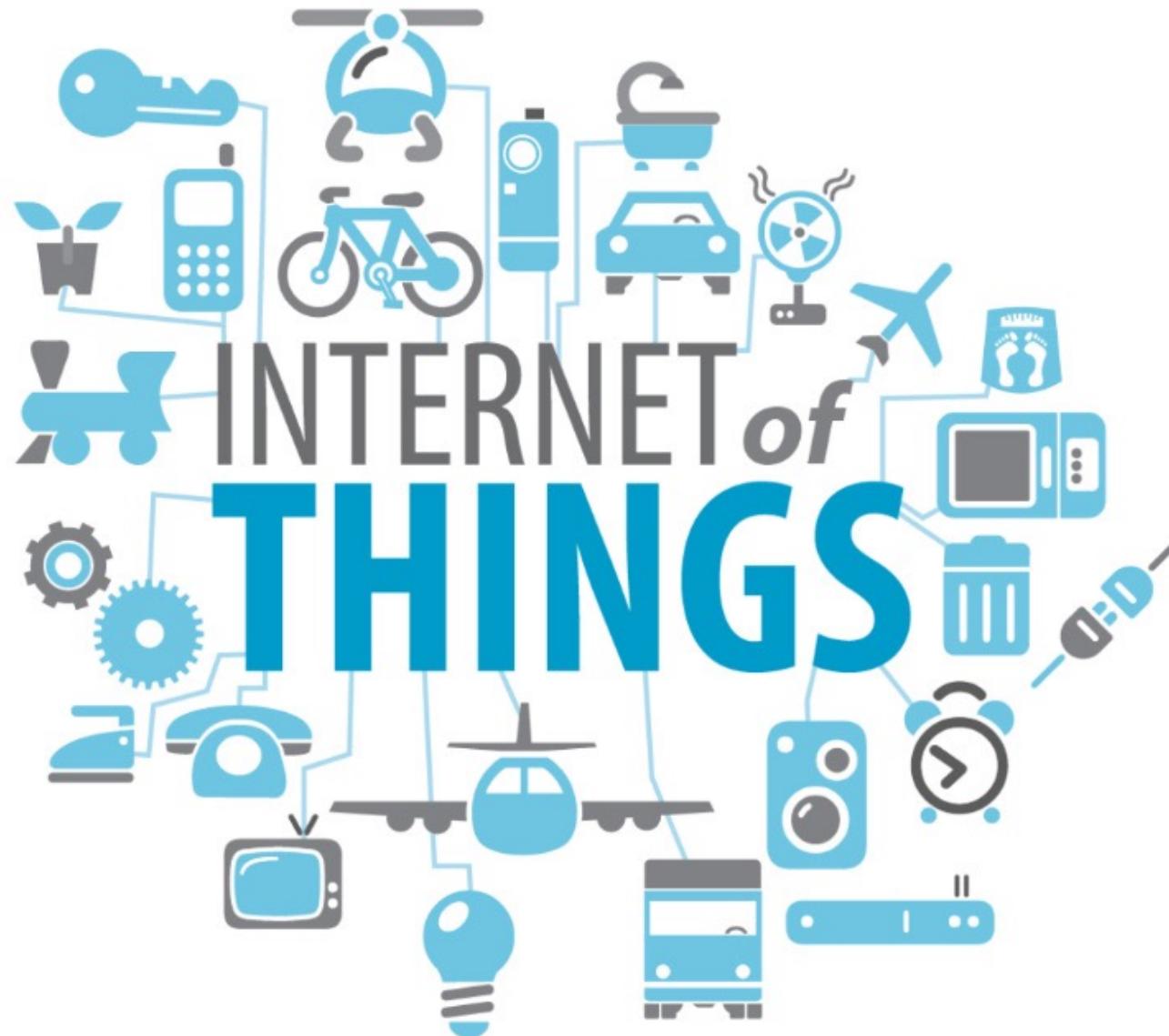


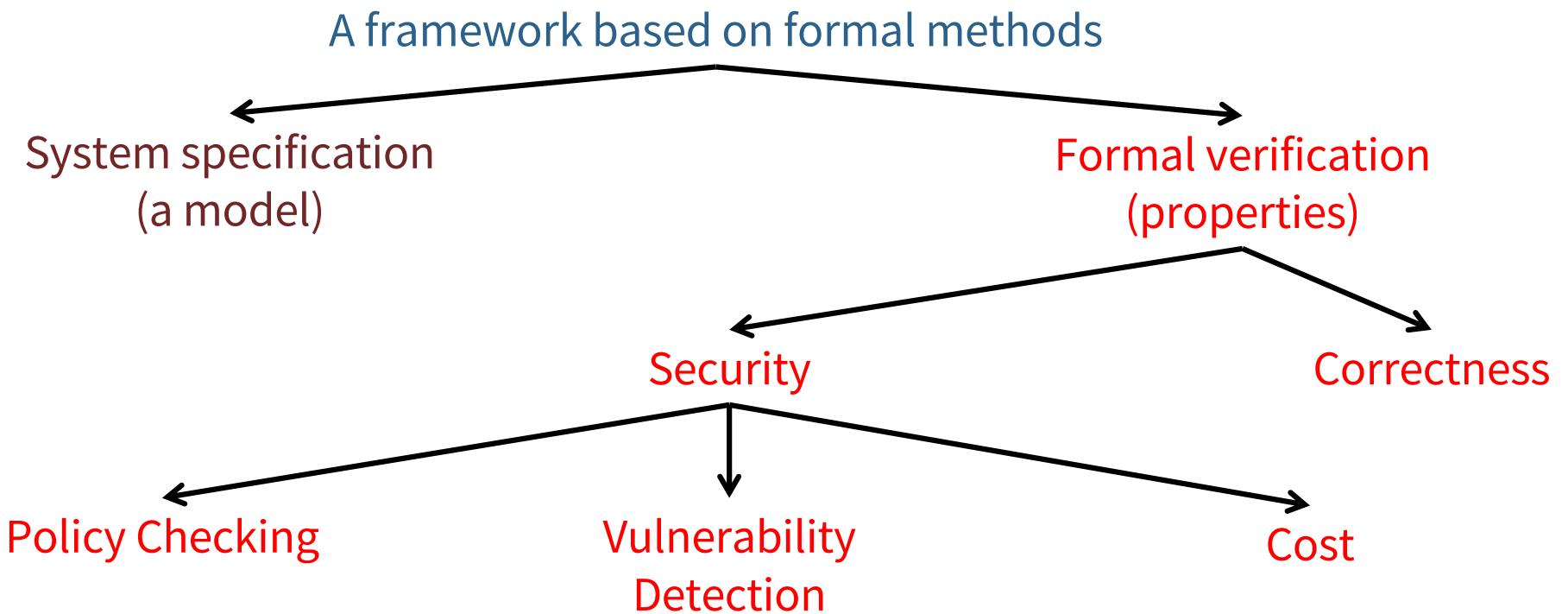


Revisiting IoT CFA for our homework





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

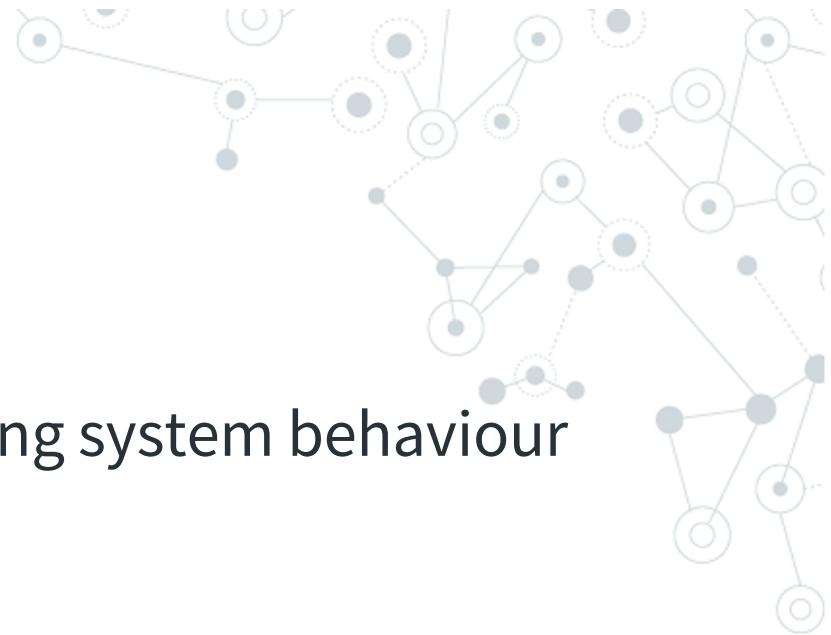


Our proposal: IoT-LySa

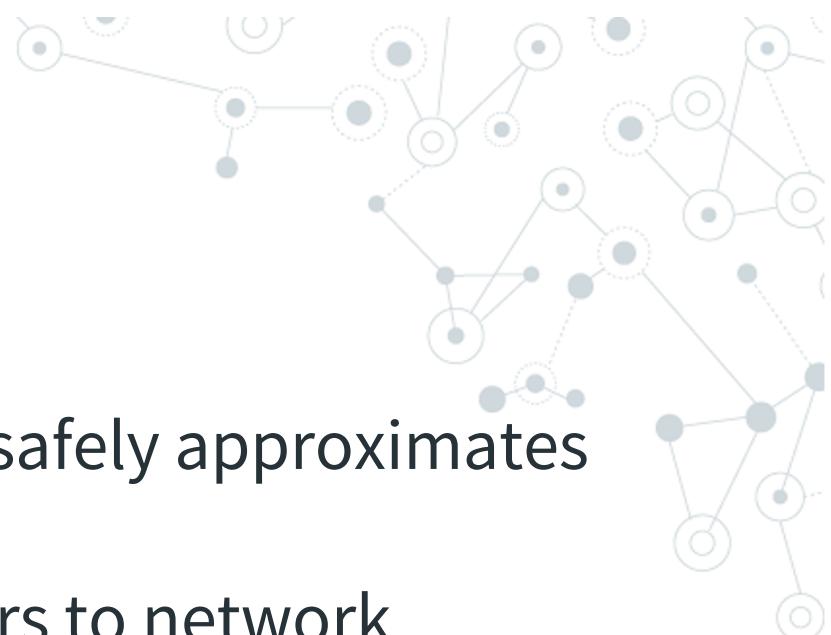
A specification language for modelling system behaviour

Constructs

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



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



Goal: Determine whether our design is robust against data manipulations

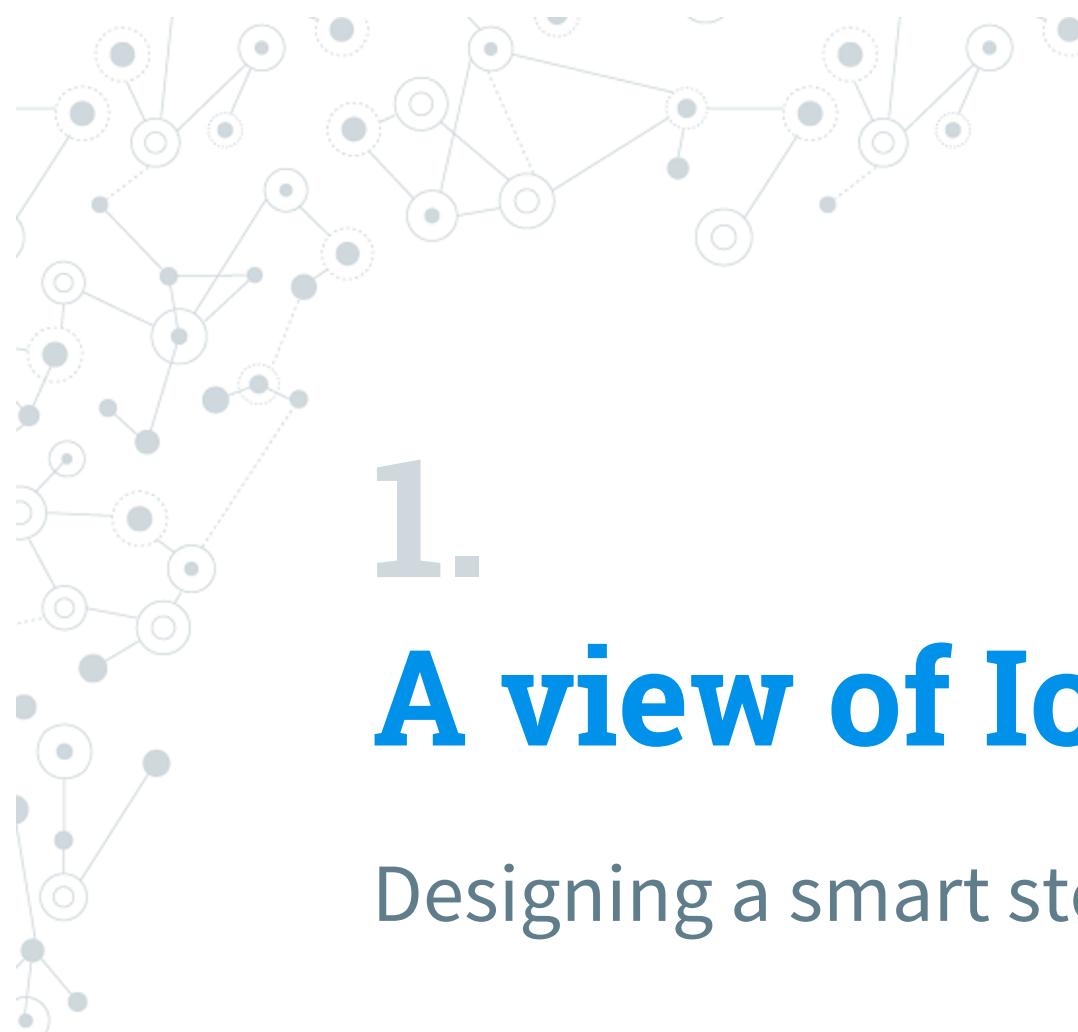
- Do critical computations depend on manipulated data?



How: Over-approximate from which data sources computations depend

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





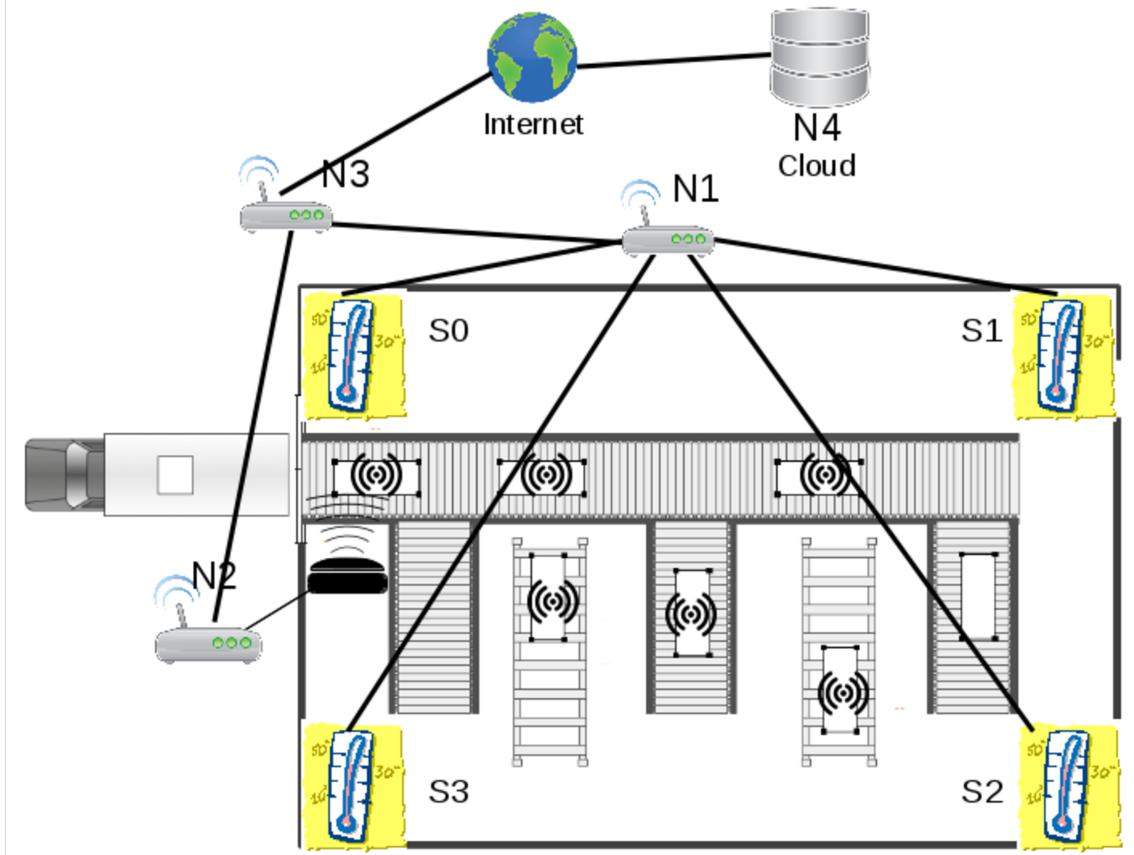
1.

A view of IoT-LySa

Designing a smart storehouse



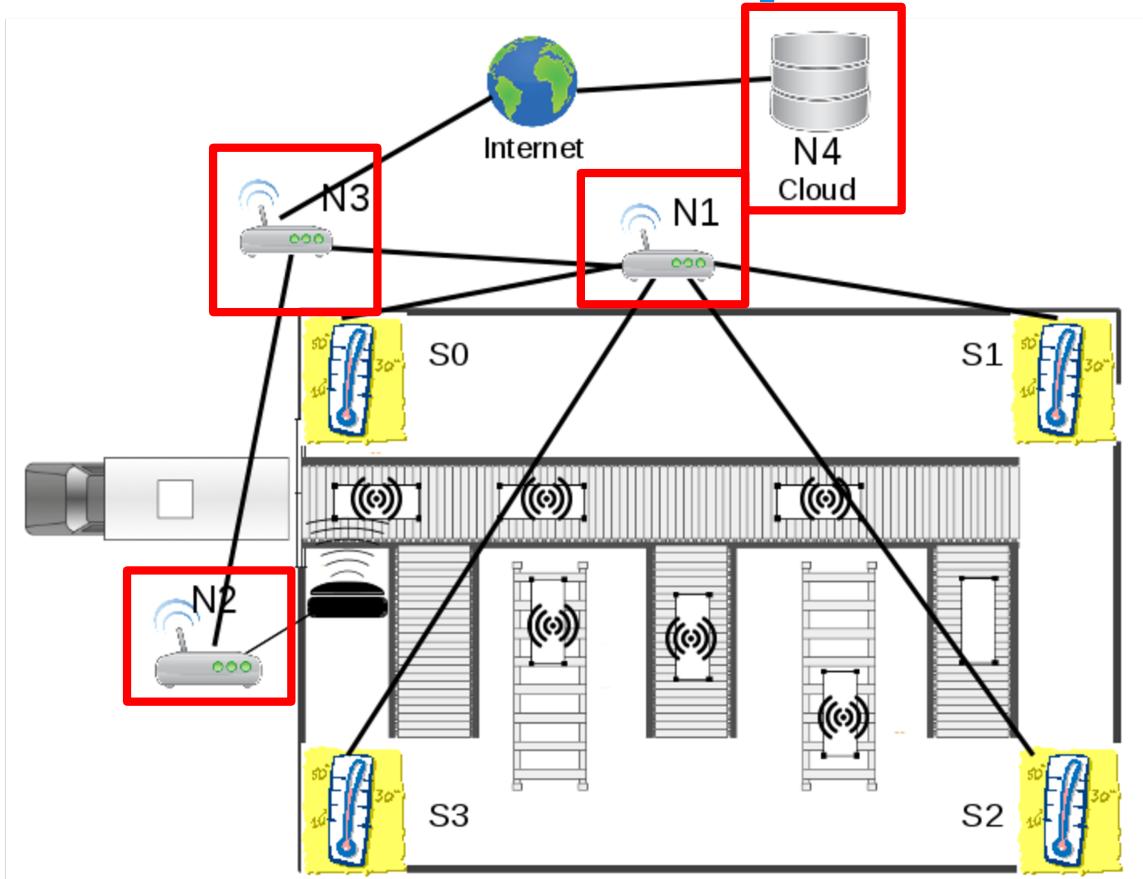
A smart storehouse with perishable food



Un magazzino che ha quattro sensori nei quattro angoli della stanza, sostanzialmente che vengono controllati da un nodo n1 , il quale raccoglie ripetutamente i dati rilevati, quindi sostanzialmente le temperature ne fa una media e controlla se questa temperatura media sta all'interno di alcuni range opportuni, quindi in particolare laddove si sta parlando di comportamento standard, quindi normale, la media serve per regolare il sistema di condizionamento e di refrigerazione ma allo stesso tempo il controllo della temperatura fa sì che si controlli se non ci sono situazioni anomale, nel qual caso la situazione viene segnalata a gli strati superiori del sistema di nodi e in qualche modo si dovrebbe poter intervenire, contemporaneamente la media dei valori di temperatura viene utilizzata dal nodo n3, quindi viene spedita dal nodo n1 al nodo n3 affinché il nodo n3 consideri se la temperatura rilevata è adeguata al carico di merce in questo momento stoccati all'interno del magazzino e anche in questo caso si prenderanno decisioni diverse a seconda della situazione.



A smart storehouse with perishable food

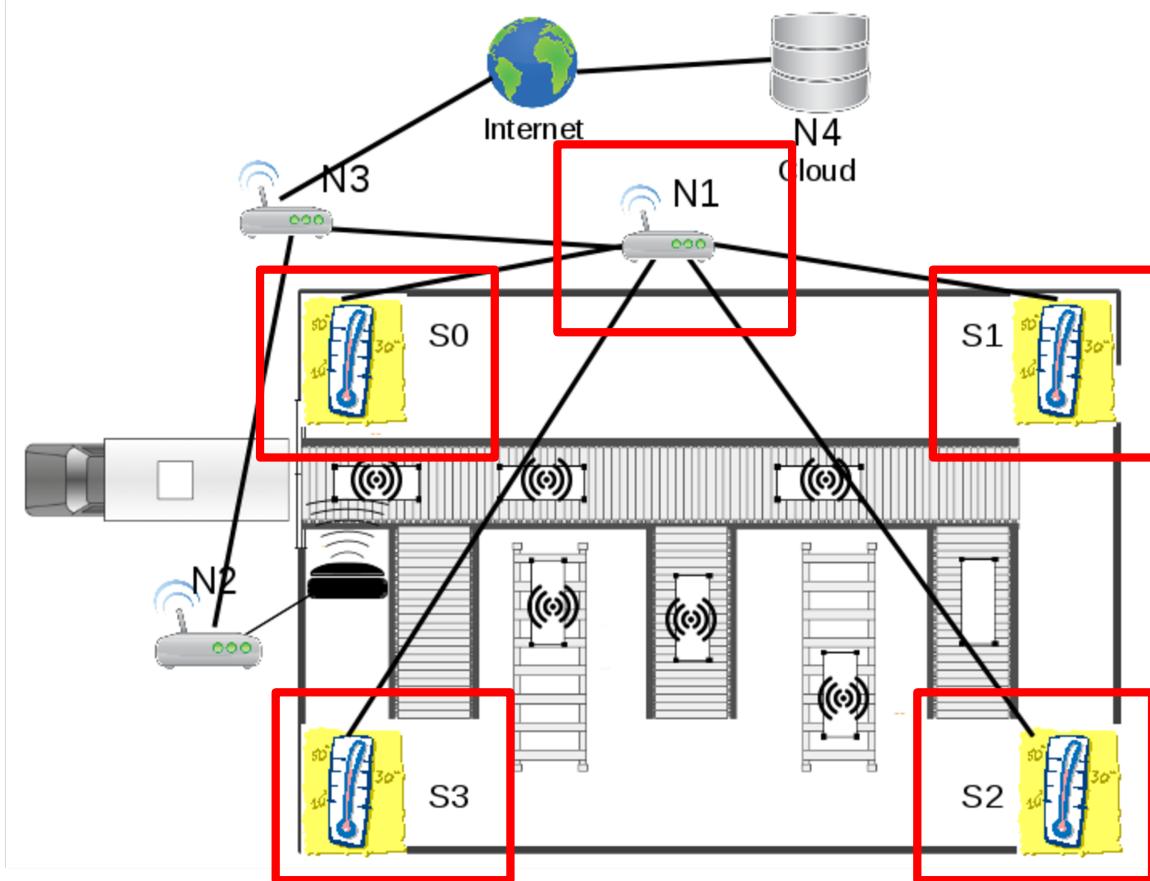


Four components:
N1, N2, N3, N4

quindi, come vedete i dati che vengono raccolti dai sensori servono sempre per prendere decisioni di attuazione che quindi possono essere critiche, perché appunto se si sbaglia a gestire il sistema di refrigerazione, quello che può succedere è che la merce è deperibile e quindi deperirà ovviamente.



A smart storehouse with perishable food

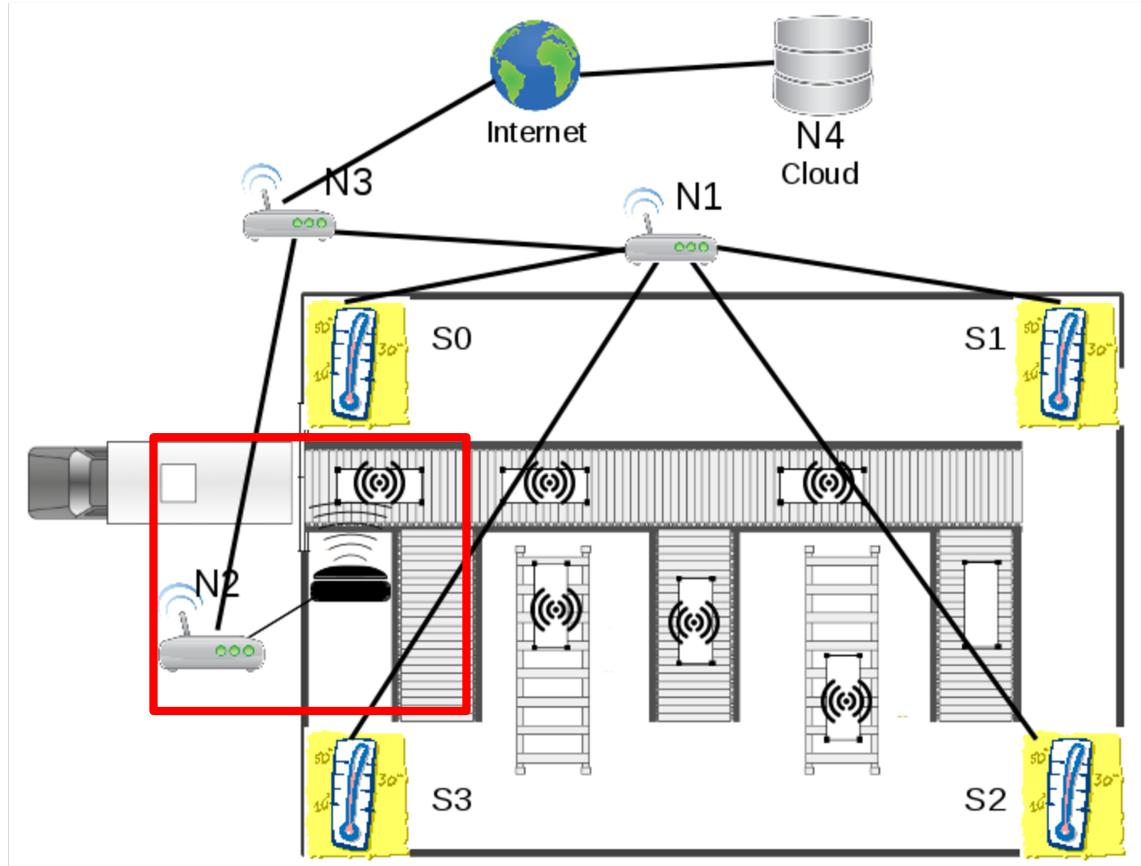


Node N1

- 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

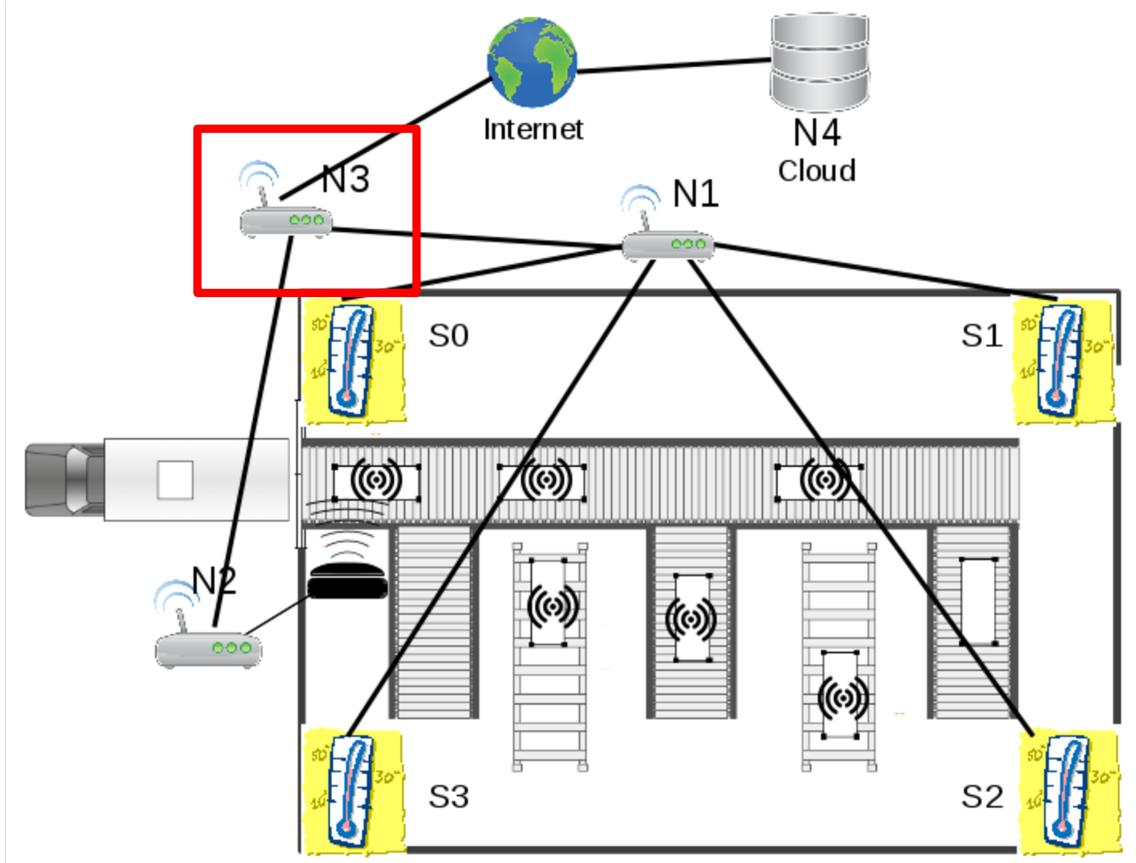


Node N2

- 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

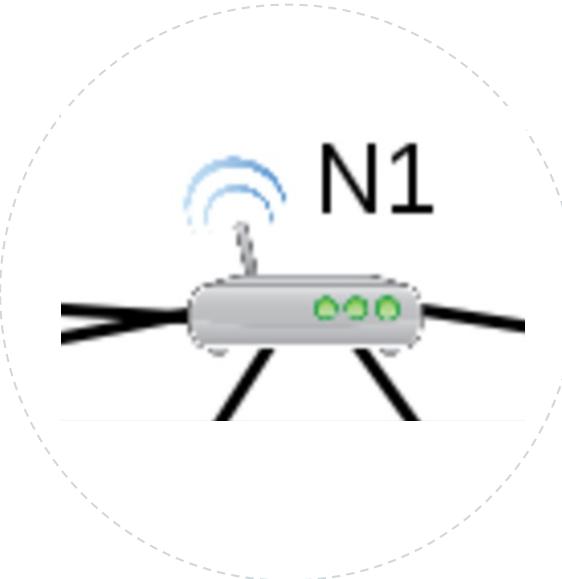


Node N3

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



IoT-LySa specification



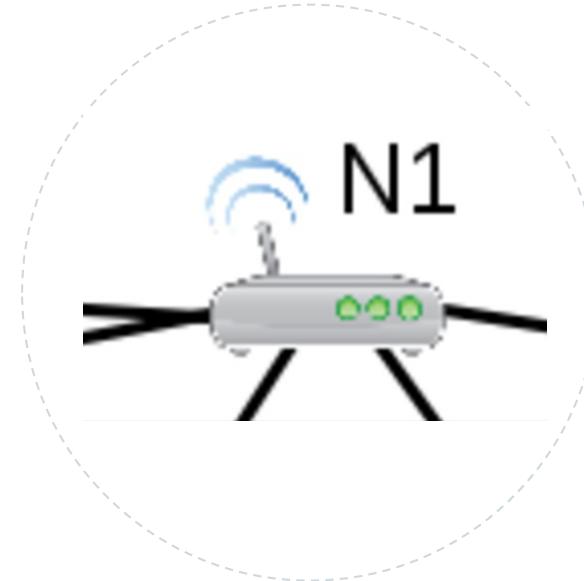
$$N_1 = \ell_1 : [\Sigma_1 \parallel P_c \parallel (S_0 \parallel S_1 \parallel S_2 \parallel S_3)]$$



IoT-LySa specification

Node id

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



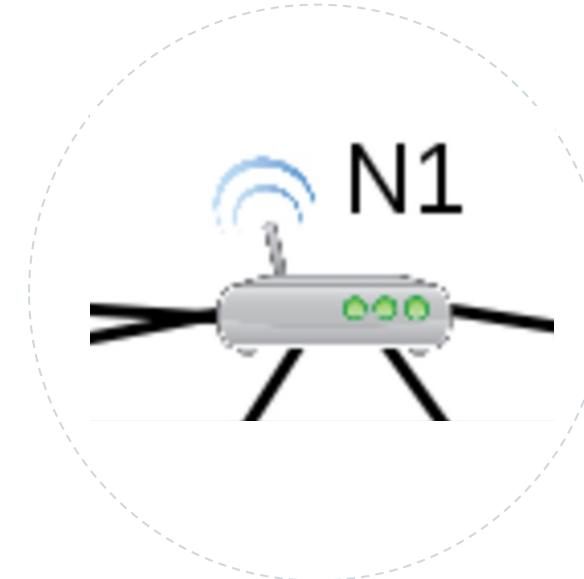
La specification di IoT LySa è sempre quella di avere un sistema di nodi, ogni nodo è etichettato,

IoT-LySa node specification

Node id

Shared store

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



ogni nodo contiene la memoria, che è il modo appunto locale per condividere informazione,

IoT-LySa node specification

Node id Shared store

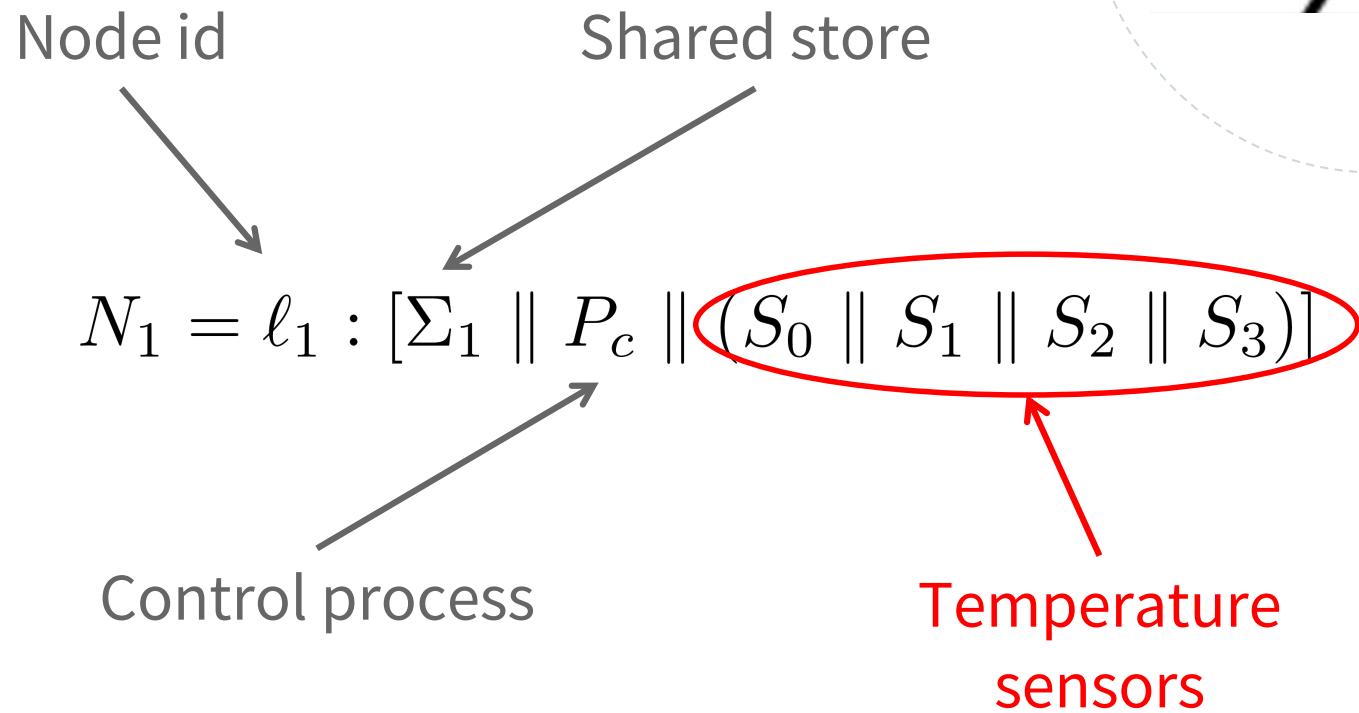
$$N_1 = \ell_1 : [\Sigma_1 \parallel \textcolor{red}{P_c} \parallel (S_0 \parallel S_1 \parallel S_2 \parallel S_3)]$$

Control process



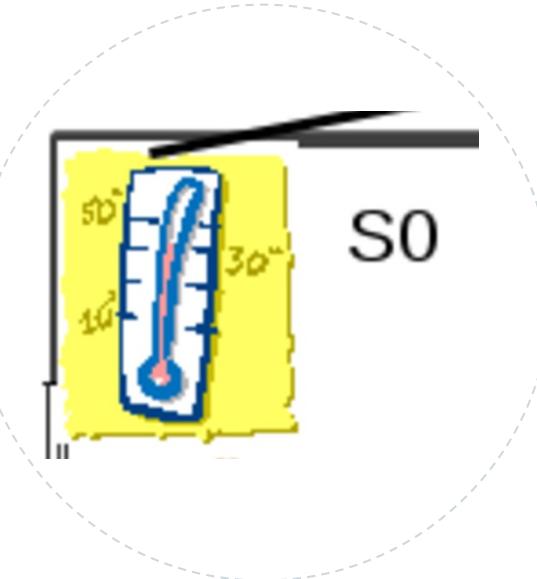
un processo o più processi in parallelo che gestiscono la logica del nodo

IoT-LySa node specification



i sensori in questo caso di temperatura i sensori si comportano come sappiamo, raccolgono la temperatura facendo il sensing di un valore

Sensor specification



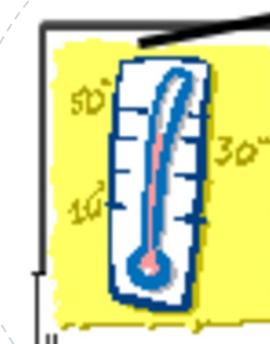
$$S_0 = \mu h. 0 := v. \tau. h$$



Sensor specification

Iteration

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

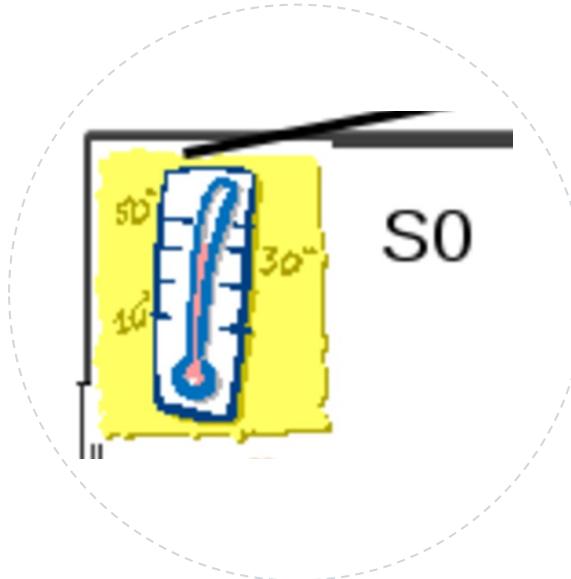


Sensor specification

Internal activities

Iteration

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



Sensor specification

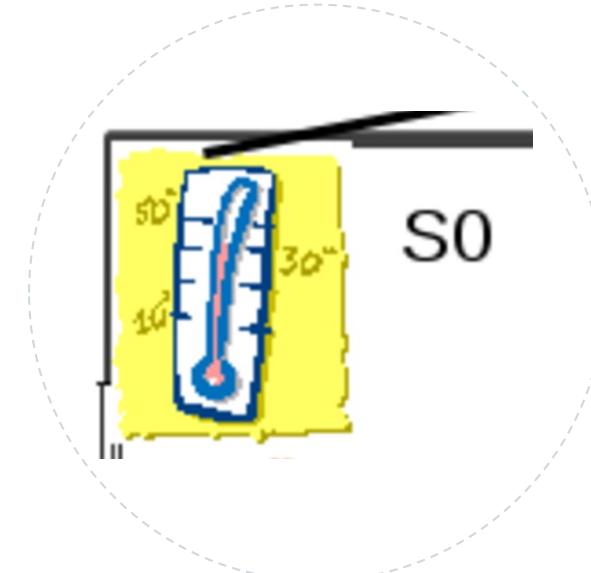
Internal activities

Iteration

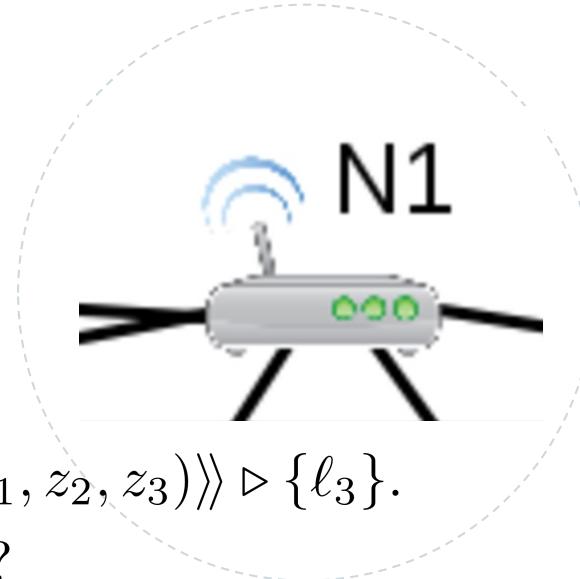
$$S_0 = \mu h. 0 := v. \tau. h$$

Sensing a value

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



Control process specification



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

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

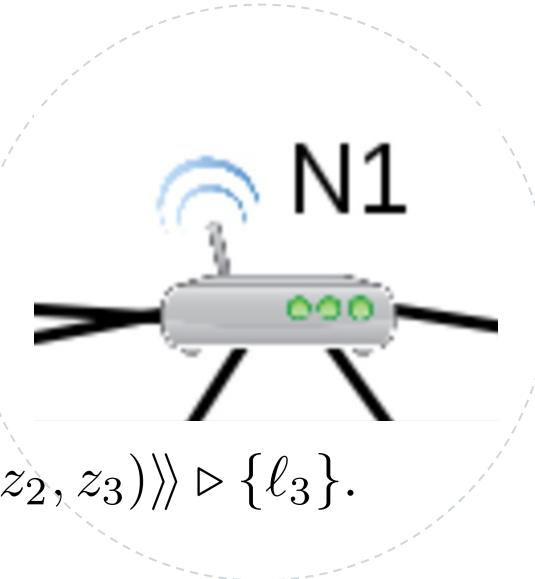
$$th_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 ? h$$

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

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

poi invece la specifica del processo di controllo è quella che appunto raccoglie i dati del sensore e fa, come dicevamo prima, i controlli sulla media se sta dentro o fuori i range, eventualmente indica all'attuatore qual è il tipo di azione che deve fare e in caso di anomalia invece segnala attraverso un messaggio di errore e poi propaga l'informazione comunque al nodo n3 affinché faccia il controllo rispetto al quantitativo di materiale stoccati all'interno del magazzino

Control process specification



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

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

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

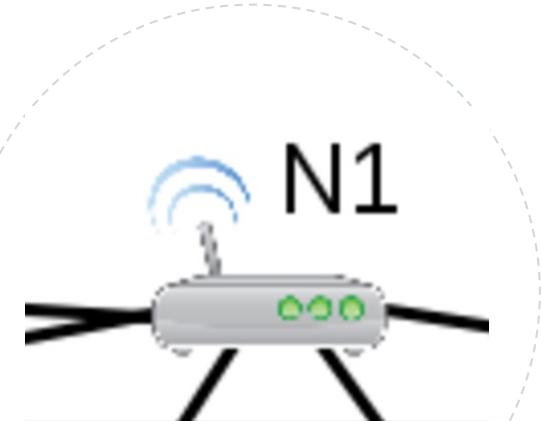
: $\langle j, \text{start} \rangle . h$

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

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

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

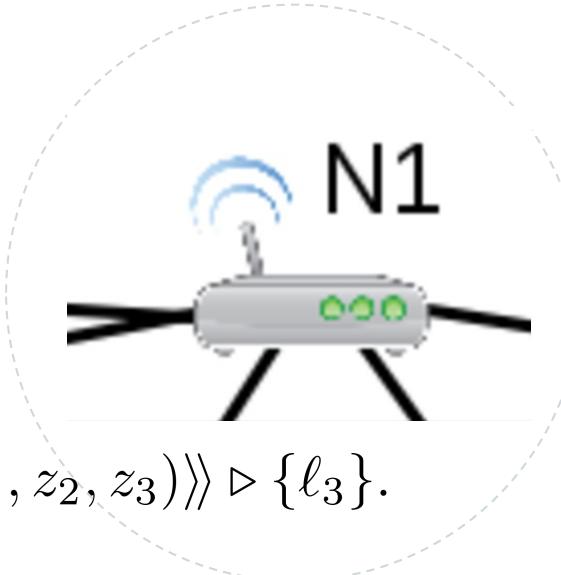
$\text{th}_1 - \text{th}_3 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq \text{th}_2 + \text{th}_3 ?$

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

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

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

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



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

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

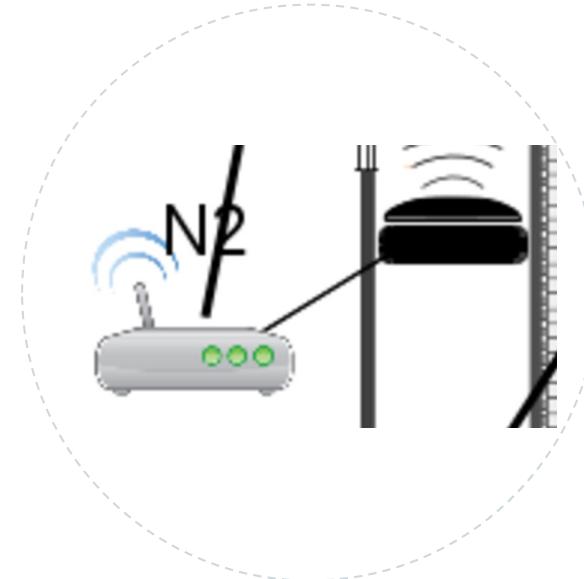
$$th_1 \leq \text{avg}(z_0, z_1, z_2, z_3) \leq th_2 ? h$$

$$\begin{array}{c} : \langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}.h \\ \xrightarrow{\hspace{10cm}} \quad : \langle j, \text{start} \rangle . h \end{array}$$

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



Other nodes: stocktaking



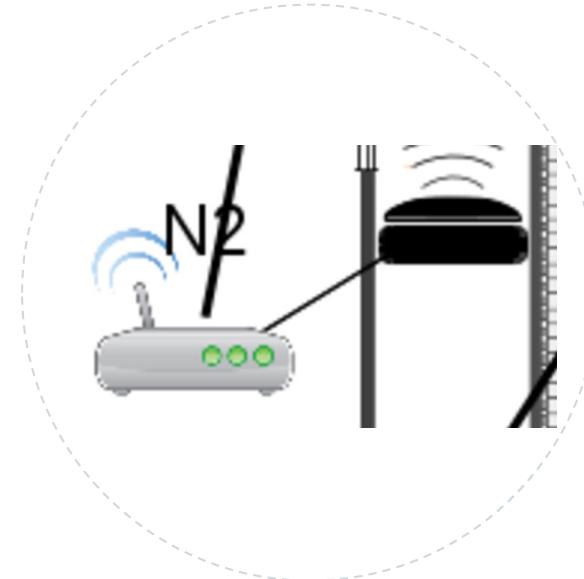
$$N_2 = \ell_2 : [\Sigma_2 \parallel R_0 \parallel \mu h. x := 0. db := update(db, x). \tau. h \parallel \mu h. \langle\langle db \rangle\rangle \triangleright \{\ell_3\}. h]$$



Other nodes: stocktaking

Shared store +
RFID Reader
(as before)

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

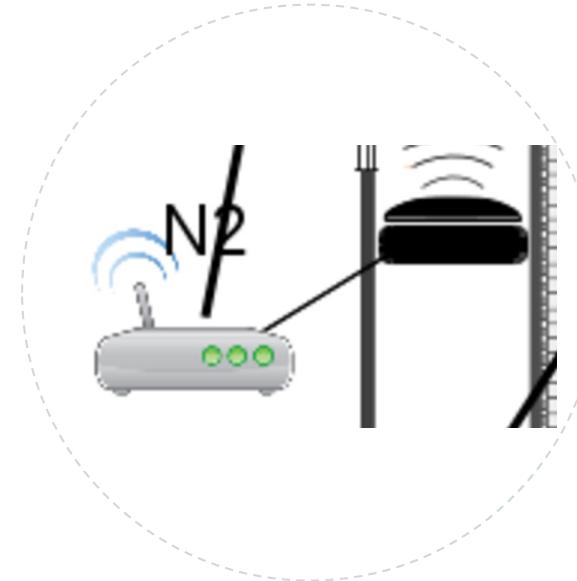


Other nodes: stocktaking

Shared store +
RFID Reader
(as before)

Update the
stocktaking with
incoming food

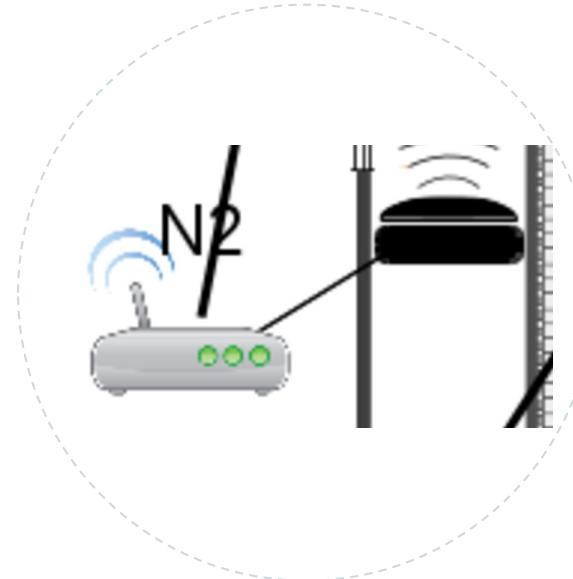
$$N_2 = \ell_2 : [\Sigma_2 \parallel R_0 \parallel \mu h. x := 0. db := update(db, x). \tau. h \parallel \\ \mu h. \langle\langle db \rangle\rangle \triangleright \{\ell_3\}. h]$$



Other nodes: stocktaking

Shared store +
RFID Reader
(as before)

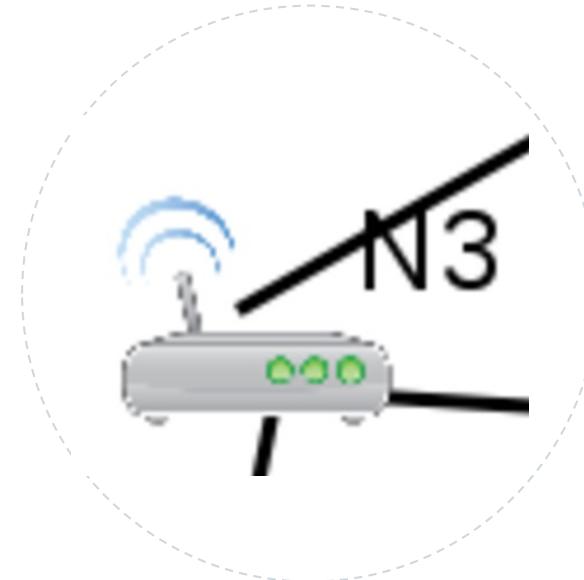
Update the
stocktaking with
incoming food


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

Send the stocktaking
to N3

questa è la parte del invece di n2 che controlla il magazzino e una volta che ha sotto controllo tutto il carico del magazzino lo spedisce a n3

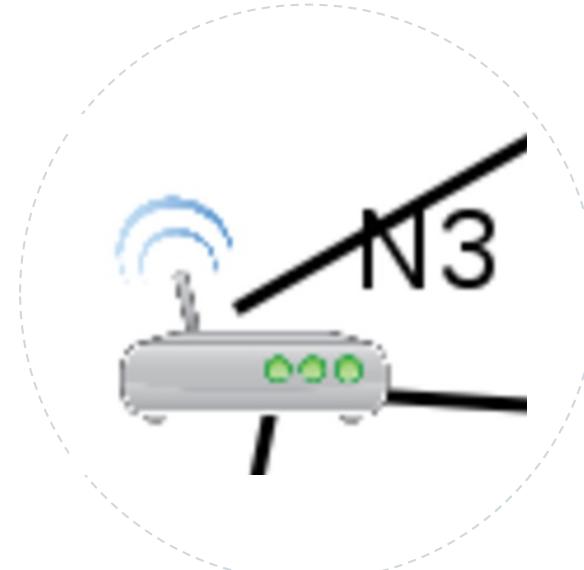
Other nodes: checking temperature



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

n3 deve andare a controllare se siamo dentro in un range valido e
altrimenti prendere provvedimenti se la temperatura non è appropriata,
allora deve segnalare più in alto che sta succedendo qualcosa ed n1 dovrà
intervenire

2.

A view of the analysis

Checking the smart storehouse

Abstract values

Trees with a finite depth d

$\hat{\mathcal{V}} \ni \hat{v} ::= \text{abstract terms}$

(\top, b) abstract value denoting cut (see below)

(ν, b) abstract value for clear data

$(f(\hat{v}_1, \dots, \hat{v}_n), b)$ abstract value for aggregated data

$(\{\hat{v}_1, \dots, \hat{v}_n\}_{k_0}, b)$ abstract value for encrypted data



And with a tag b

\diamond untainted

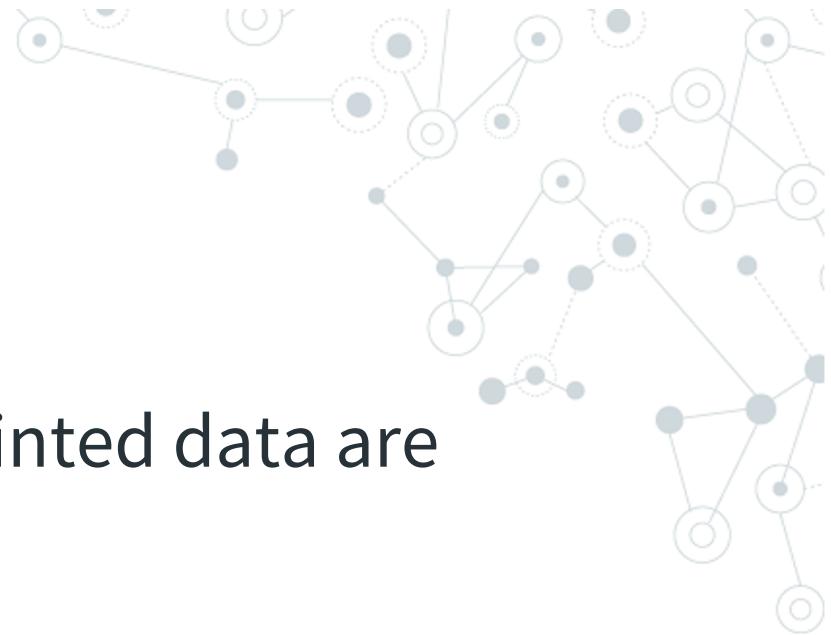
\diamond tamperable

\diamond sensitive

\diamond 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

l'analisi è basata sempre su quella che abbiamo visto per IoT LySa con una piccola variazione sul tema, ovvero i valori astratti, abbiamo detto che la nostra analisi non tiene conto dei valori concreti perché quello che interessa è tracciare la provenienza dei dati, quindi tracciare il flusso dei dati indipendentemente dal loro valore concreto quindi non ci interessa sapere se la temperatura è 30 o 35 o 25, ma ci interessa sapere che parte dal sensore e in questa particolare analisi basta sapere la natura di questa rilevazione, se proviene da una fonte attendibile oppure no e quindi associamo al valore astratto, per esempio v un altro elemento nella coppia che è b, il tag che ci dice se l'informazione è affidabile o meno, o sensitive o meno quindi abbiamo queste quattro possibilità, il quadratino nero vuol dire che non c'è nessun tipo di informazione di cui tenere conto ,il quadratino rosso con la linea verticale vuol dire che l'informazione può essere tamperable, devo stare attento anche se il valore è taggato etichettato come sensitive con il quadratino azzurro e la linea orizzontale e naturalmente si può avere anche la combinazione delle due cose nell'ultimo possibile tag b che è il quadratino fucsia che ha la barra sia orizzontale che verticale, che sta proprio per la combinazione delle due cose. Naturalmente questo è quello che succede in partenza, però questo tipo di informazioni deve essere propagato man mano che la comunicazione all'interno del sistema di nodi propaga i valori concreti che corrispondono ai valori astratti che abbiamo appena visto. Quindi dovremmo vedere e poi lo vedremo meglio tra brevissimo è come viene fatta la propagazione al livello di analisi, quindi l'idea è che i valori si portino dietro il loro pezzettino di taint information e lo propaghino opportunamente mano a mano che vengono inseriti in tuple che vengono mandate a loro volta in aggregazione, eccetera e naturalmente questo è un sistema in cui vengono proposte alcune policy di propagazione, ma anche naturalmente sono esempi, possono essere raffinate in altri modi e addirittura fornite dagli stessi progettatori che hanno in testa che cosa serve loro.

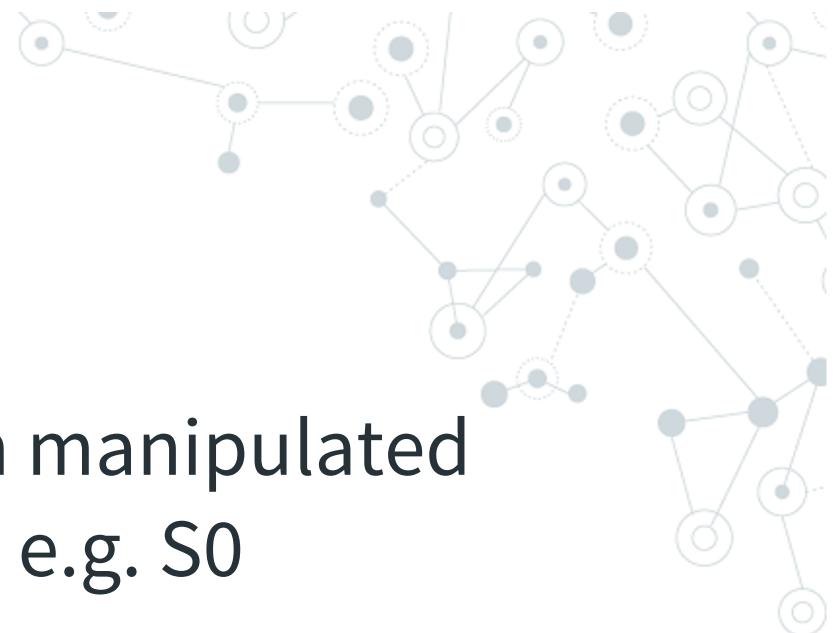


Using static taint analysis: ingredients

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

L'analisi in particolare classificherà le sorgenti di dati in modo da tener conto se sono tainted oppure no, determinerà i punti critici della computazione laddove tutte queste informazioni arrivano e vengono utilizzate per prendere decisioni e poi potremmo definire qualche esempio di policy di propagation.

Static taint analysis in our example



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

Its values are tagged with \diamond

2. The critical point is the test in N3

$$(temp \notin validRange(db))^a$$

3. Default taint propagation policy

nel nostro esempio, ad esempio, possiamo ipotizzare che un attaccante sia in grado di manipolare solo un sensore, ad esempio il sensore s0, quindi uno può immaginare che fisicamente nel magazzino ci sia un'area aperta, cioè meno protetta cui può accedere un malintenzionato e manomettere il sensore e nel caso sempre del nostro esempio, quindi ho bisogno di taggare i valori che provengono dal sensore con il tag di tipo rosso ora l'abbiamo già visto più o meno raccontando il case study, vediamo che un punto critico di decisione è quello in cui in n3 si va a vedere se la temperatura è adeguata al range necessario per mantenere refrigerata la merce stoccatà e quindi è chiaro che io devo andare a controllare quanto possa essere l'impatto di quel sensore che da un valore possibilmente non preciso alla decisione.

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
$\hat{\Sigma}$	◇	◇	◇	◇	◇			



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
$\hat{\Sigma}$	◇	◇	◇	◇	◇	◇	◇	◇

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



A questo punto, siccome l'informazione raccolta dal processo va a formare poi il valore medio della temperatura, è chiaro che dato che la media considera anche z_0 anche la media sarà in qualche modo macchiata da questa componente e quindi in questa maniera vi potete immaginare intuitivamente come l'informazione si propaga ogni volta che io utilizzo un'informazione in un contesto di una tupla oppure nel contesto di una funzione o di un encryption e laddove io mi porto dietro un'informazione che prima era stata considerata taint, sto inquinando anche l'aggregazione successiva.

Analysis of N1



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

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

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

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

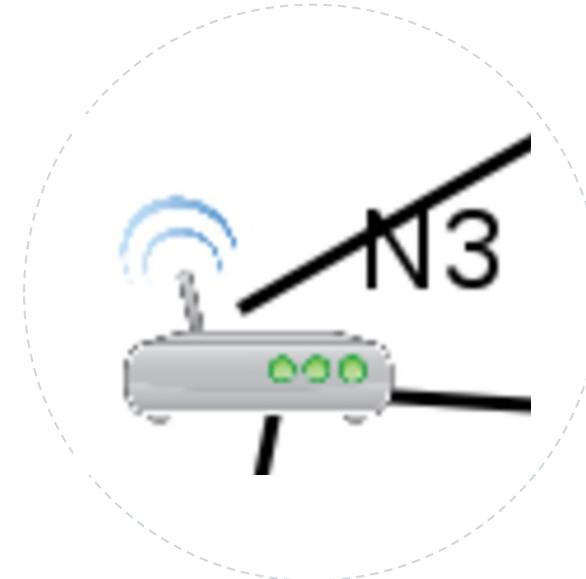
: $\langle\langle \text{alarm} \rangle\rangle \triangleright \{\ell_3\}.h$

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

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



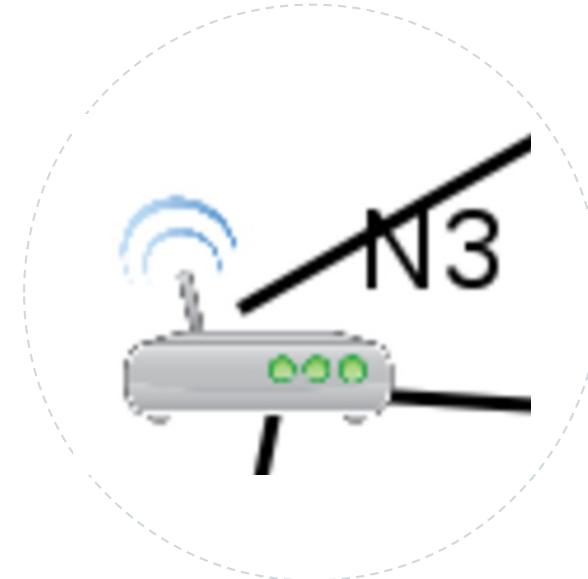
Analysis of N3


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

: h

La nostra analisi applicata al punto critico che abbiamo identificato qui con A ci dice che la decisione dipende anche da un qualcosa che potenzialmente è temparable e quindi ci crea la situazione in cui dobbiamo portare attenzione.

Analysis of N3



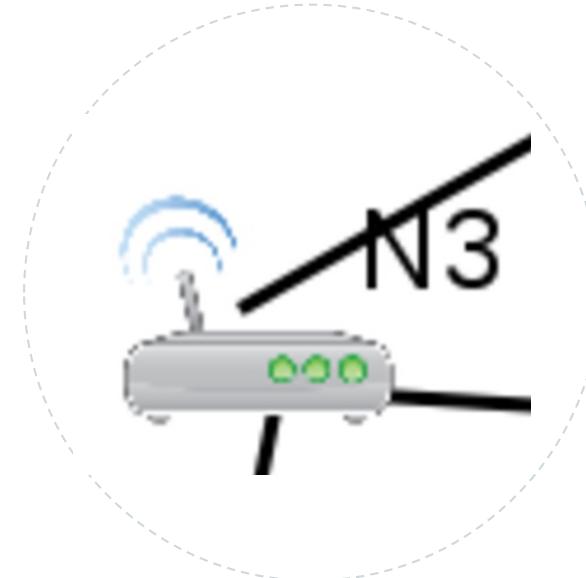
$\mu h.(temp \notin validRange(db))^a ? \langle\langle \text{outRange} \rangle\rangle \triangleright \{\ell_4\}.$
 $\langle\langle validRange(db) \rangle\rangle \triangleright \{\ell_1\}.h$

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

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



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



More in details



Abstract values

Più nel dettaglio rivediamo che abbiamo quindi tag che arricchiscono i valori astratti e quindi possiamo scriverli anche nella forma v alla b se non volete la coppia e vediamo come l'analisi sostanzialmente ne tiene conto.

Trees with a finite depth d



$\hat{\mathcal{V}} \ni \hat{v} ::= \text{abstract terms}$

(T, b)	abstract value denoting cut (see below)
(ν, b)	abstract value for clear data
$(f(\hat{v}_1, \dots, \hat{v}_n), b)$	abstract value for aggregated data
$(\{\hat{v}_1, \dots, \hat{v}_n\}_{k_0}, b)$	abstract value for encrypted data

And with a tag b (we use v^b)

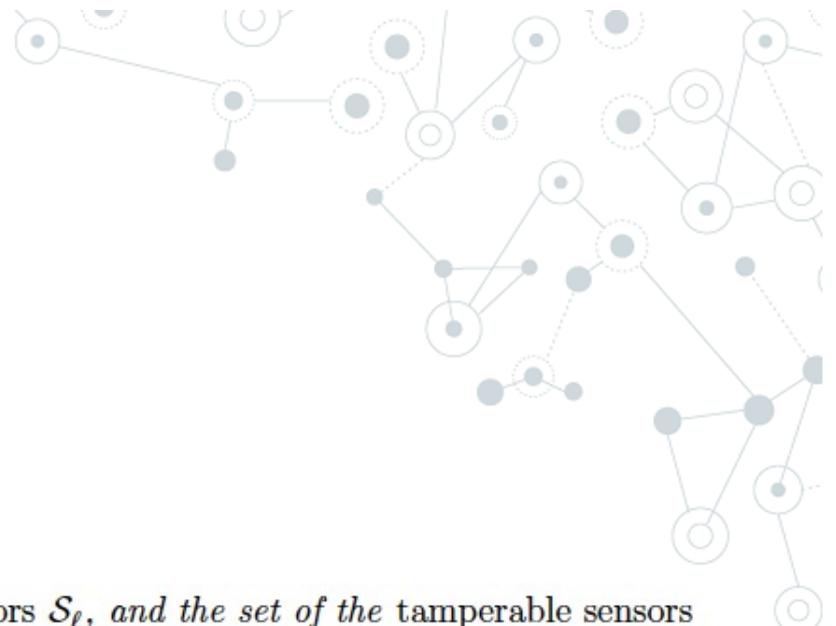
◊ untainted

❖ tamperable

❖ sensitive

❖ tamperable + sensitive

Dal punto di vista dello sviluppo formale, quindi io ho deciso come chiamare queste etichette, adesso devo decidere come fare in qualche modo a legarle ai pezzettini della nostra estimate o soluzione, quindi io ho bisogno di intanto capire quali sono le variabili su cui impatta la natura taint delle informazioni e vedere poi sostanzialmente come queste vengono propagate.



Data source classification

Definition 2.1 (Data classification). *Given the set of sensitive sensors S_ℓ , and the set of the tamperable sensors and variables T_ℓ , the taint assignment function τ is defined as follows:*

$$\tau(y, \ell) = \begin{cases} \diamond & \text{if } y \in S_\ell \\ \diamond & \text{if } y \in T_\ell \\ \diamond & \text{if } y \in S_\ell \cap T_\ell \\ \diamond & \text{o.w.} \end{cases} \quad \text{where } y \in I_\ell \cup \mathcal{X} \quad \tau(v, \ell) = \diamond$$

Per quanto riguarda la classificazione dei dati io ho bisogno quindi come prima cosa di legare alle variabili che utilizzo all'interno del mio sistema il tag di tipo taint, quindi vediamo semplicemente che si può usare una funzione di assegnamento una funzione τ , che si applica a una coppia il cui primo elemento è la variabile, il secondo è l'etichetta del nodo e basandomi su una classificazione rispetto ai nodi di quelli che sono i sensori di tipo sensitive o i sensori di tipo tamperable, io vado a associare di conseguenza il tag, quindi se ho una variabile y che fa parte delle variabili di un sensore che è considerato sensitive allora io vado a segnarmi l'informazione all'interno e quindi la lego alla variabile, viceversa o analogamente se se y è in quota sia nell'insieme dei sensori sensitive che di quelli tamperable metto il solito tag congiunto, altrimenti vuol dire che non succede niente, quindi vado a mettere semplicemente il tag che dice che non ci sono taint di sorta. A questo punto dobbiamo decidere quali sono le policy di propagazione ora, come dicevamo la volta scorsa, dobbiamo naturalmente dividere la propagazione dei sensitive da quella dei tamperable perché la propagazione dei sensitive segue un po la logica che abbiamo visto con anche le analisi precedenti, cioè io mi aspetto che naturalmente i valori sensitive rimangano tali in propagazione con l'unica eccezione che è quella dell'encryption, cioè se io un valore sensitive lo codifco all'interno magari di una tupla e con un encryption allora l'encryption può passare in chiaro perché come se si opacizzasse quindi ciò che è sensitive all'interno non deve essere visto e quindi mi posso immaginare un tipo di propagazione di questo genere oppure se lo pensate sul lato della privacy potete supporre, per esempio, di anonimizzare alcune informazioni, se pensate all'esempio dei lampioni e della macchina che veniva fotografata, potete pensare che l'informazione potrebbe girare con una qualche forma di blurring per non identificare, per esempio, i volti dei passeggeri, quindi uno può pensare che alcune informazioni nascono in un certo modo, ma poi si possa intervenire con delle funzioni per o annullare l'effetto o comunque contenerlo quindi uno può anche pensare a forme più graduali di propagazione. Per quanto riguarda il tamperable di nuovo uno può pensare che quello invece rimane macchiato, cioè quindi macchia una qualcosa, macchia tutto quello che tocca propagandosi a meno che anche in questo caso non ipotizzi una qualche funzione di sanitization che in qualche modo resetti la bontà del dato raccolto.

Default taint propagation policy

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



Propagation of taint information: the combination operator 1



\otimes works as a join operator

\otimes	\diamond	$\diamond\circlearrowleft$	$\diamond\circlearrowright$	$\diamond\oplus$
\diamond	\diamond	$\diamond\circlearrowleft$	$\diamond\circlearrowright$	$\diamond\oplus$
$\diamond\circlearrowleft$	$\diamond\circlearrowleft$	$\diamond\circlearrowleft$	$\diamond\circlearrowright$	$\diamond\oplus$
$\diamond\circlearrowright$	$\diamond\circlearrowright$	$\diamond\oplus$	$\diamond\circlearrowright$	$\diamond\oplus$
$\diamond\oplus$	$\diamond\oplus$	$\diamond\oplus$	$\diamond\oplus$	$\diamond\oplus$

This operator naturally extends to abstract values

Immaginiamo di avere un operatore di combinazione questa è la versione più semplice che insomma lavora sostanzialmente come un operatore di join e che si estende naturalmente anche a valori astratti ed è il combinatore che uno si aspetta, quindi la combinazione di ciò che è nero diventa nero, quando le coppie sono, tranne un'eccezione che ora vedremo, l'idea è che coppie di valori omologhi danno lo stesso valore e adesso l'unica cosa che va guardata è la combinazione dei tamperable perché ci sono dei casi in cui devo aggiungere naturalmente la combinazione, quindi come vedete nero con nero da nero, blu con blu da blu, viola con viola da viola quindi le uniche eccezioni sono ma sono attese è che se io ho da una parte un valore sensitive e dall'altra devo combinarlo con un valore di tipo tamperable, naturalmente devo mettere la combinazione dei due, ugualmente faccio nel caso simmetrico.

Propagation of taint information on functions and encryptions



Definition 2.2 (Taint propagation policies). *Given the combination operator $\otimes : \mathcal{B} \times \mathcal{B} \rightarrow \mathcal{B}$, the taint resulting by the application of*

- a function f is $F_\tau(f, \hat{v}_1, \dots, \hat{v}_r) = \otimes(\hat{v}_{1\downarrow_2}, \dots, \hat{v}_{r\downarrow_2})$
- an encryption function is $Enc_\tau(\hat{v}_1, \dots, \hat{v}_r) = \begin{cases} \diamond & \text{if } \forall i. \hat{v}_{i\downarrow_2} \in \{\diamond, \oplus\} \\ \lozenge & \text{o.w.} \end{cases}$

It is obviously possible to consider different policies for propagation, e.g. we could consider a set of functions that produce sensitive data independently from the taint information of its arguments. Moreover, we could deal with anonymisation functions that process their arguments to remove sensitive information. Consider e.g. blurring the faces of people in surveillance videos to protect their privacy. A policy for these functions is similar to the one for the encryption: the resulting taint label is \diamond , if no argument is also tamperable, otherwise is \lozenge .

A questo punto vediamo come si fa a definire una politica di propagazione proprio in termini formali, l'idea è quella che abbiamo discusso prima, quindi io suppongo di avere un operatore di combinazione e a quel punto ogni volta che io trovo una funzione vado per capire qual è il tag di tipo taint vado a fare semplicemente la combinazione dei valori astratti che mi trovo all'interno. Se supponiamo che ho due valori su 4 di tipo taint quello che succede è che la funzione diventa di tipo taint e tamperable e invece, come dicevamo prima, se sono in presenza di una funzione di encryption, allora mantengo l'informazione, per quanto dal punto di vista della secrecy vado a ricondurre a non pericoloso e non sensitive quando ho l'encryption di qualcosa di sensibile, quindi quando una delle componenti è blu mi diventa bianca proprio per effetto della protezione opacizzante, se volete, dell' encryption e questa è, come dicevamo, il tipo di politica più semplice che ci può venire naturalmente in mente, però con tutte le varianti che si stavano dicendo prima.

Propagation of taint information: the combination operator 2



\otimes	\diamond	\diamond_L	\diamond_L	\diamond_L
\diamond	\diamond_L	\diamond_L	\diamond_L	\diamond_L
$\diamond_{L'}$	$\diamond_{L'}$	$\diamond_{L \cap L'}$	$\diamond_{L \cap L'}$	$\diamond_{L \cap L'}$
$\diamond_{L'}$	$\diamond_{L'}$	$\diamond_{L \cap L'}$	$\diamond_{L \cap L'}$	$\diamond_{L \cap L'}$
$\diamond_{L'}$	$\diamond_{L'}$	$\diamond_{L \cap L'}$	$\diamond_{L \cap L'}$	$\diamond_{L \cap L'}$

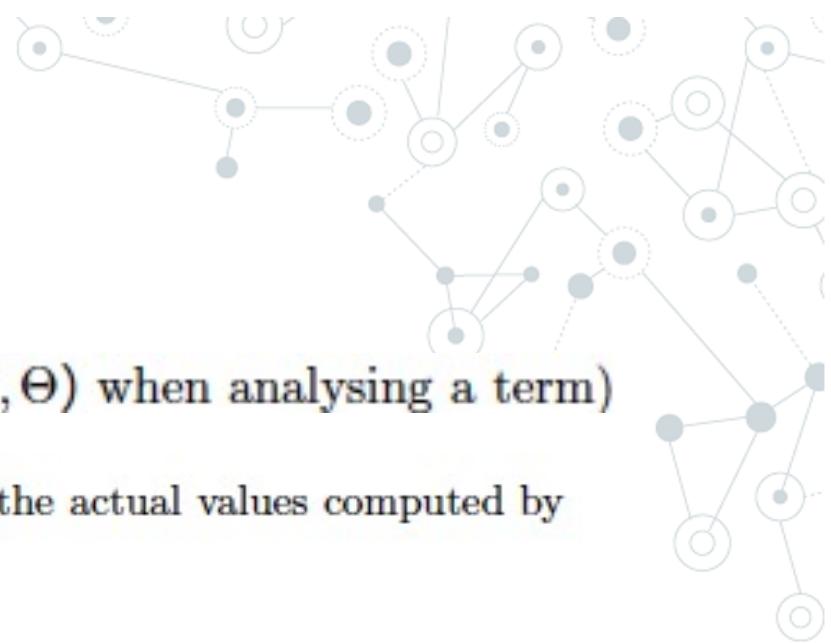
Finally, we could extend the whole presented schema to keep also the source nodes of tainted information. To do that, we could introduce in our abstract values also the labels of nodes from which the taint information derives. To deal with this new information, the taint assignment function τ returns pairs (b, L) , written b_L (b_ℓ when $L = \{\ell\}$), our combinator operator becomes the one shown on the right part in Table 1, and the encryption operator takes ℓ as argument and uses it to annotate the resulting taint label.

un'altra possibilità è più fine e da più informazioni è quella di mantenere più o meno la combinazione che abbiamo visto, aggiungendo informazioni locali, cioè possiamo tenere conto e quindi in qualche modo registrare nell'analisi anche i nodi sorgenti dell'informazione taint e quindi potremmo introdurre dei valori astratti che tengono conto anche di questa informazione, quindi sostanzialmente che introducono b ed l , non solo b , in modo tale che ogni volta che vado a combinare mi porto anche dietro l'intersezione degli insiemi di etichette che sono quelli dei nodi coinvolti. Questo dà qualche informazione in più, questo ovviamente tutte le volte che uno fa operazioni di questo genere deve capire se questo tipo di informazioni può essere utile ai fini poi dell'applicazione ovviamente.

CFA

The analysis result is a triple $(\hat{\Sigma}, \kappa, \Theta)$ (a pair $(\hat{\Sigma}, \Theta)$ when analysing a term)

a super-set $\Theta : \mathcal{L} \rightarrow \mathcal{A} \rightarrow 2^{\mathcal{P}}$ of the taint information of the actual values computed by each labelled term M^a in a given node ℓ , at run time.



The judgements for labelled terms have the form $(\hat{\Sigma}, \Theta) \models_{\ell} M^a$

$$\frac{\tau(x, \ell) \otimes \hat{\Sigma}_{\ell}(x) \subseteq \Theta(\ell)(a)}{(\hat{\Sigma}, \Theta) \models_{\ell} x^a}$$

This combination allows us to propagate the tamperable taint if x is taint



CFA (cont.)

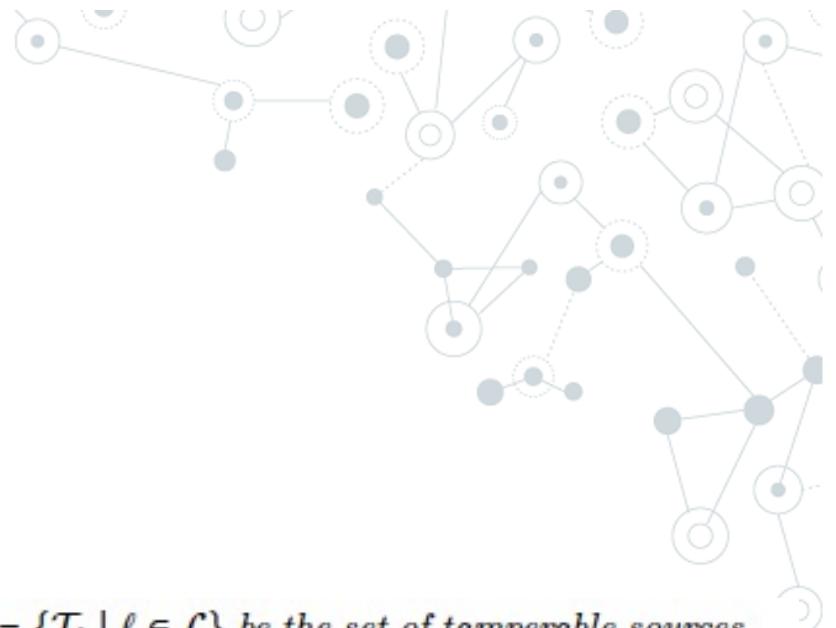


$$\begin{array}{c}
 \frac{\bigwedge_{i=1}^k (\hat{\Sigma}, \Theta) \models_{\ell} M_i^{a_i} \wedge (\hat{\Sigma}, \kappa, \Theta) \models_{\ell} P \wedge \\
 \forall \hat{v}_1, \dots, \hat{v}_r : \bigwedge_{i=1}^r \hat{v}_i \in \Theta(\ell)(a_i) \Rightarrow \forall \ell' \in L : (\ell, \langle\langle \hat{v}_1, \dots, \hat{v}_r \rangle\rangle) \in \kappa(\ell')}{(\hat{\Sigma}, \kappa, \Theta) \models_{\ell} \langle\langle M_1^{a_1}, \dots, M_r^{a_r} \rangle\rangle \triangleright L.P} \\
 \\
 \frac{\forall (\ell', \langle\langle \hat{v}_1, \dots, \hat{v}_r \rangle\rangle) \in \kappa(\ell) : Comp(\ell', \ell) \Rightarrow \left(\bigwedge_{i=j+1}^r \hat{v}_i \otimes \tau(x_i, \ell) \in \Sigma_{\ell}(x_i) \wedge (\hat{\Sigma}, \kappa, \Theta) \models_{\ell} P \right)}{(\hat{\Sigma}, \kappa, \Theta) \models_{\ell} (M_1^{a_1}, \dots, M_j^{a_j}; x_{j+1}^{a_{j+1}}, \dots, x_r^{a_r}).P}
 \end{array}$$

Our analysis respects the operational semantics of IoT-LYSA.

In the following, we denote with $N \xrightarrow{\overline{M_1^{a_1}, \dots, M_r^{a_r}}}_{\ell} N'$ when all the terms $M_i^{a_i}$ are evaluated inside node ℓ , and with $N \xrightarrow{\langle\langle v_1, \dots, v_r \rangle\rangle}_{\ell_1, \ell_2} N'$ when the message $\langle\langle v_1, \dots, v_r \rangle\rangle$ is sent from the node ℓ_1 to the node ℓ_2 .



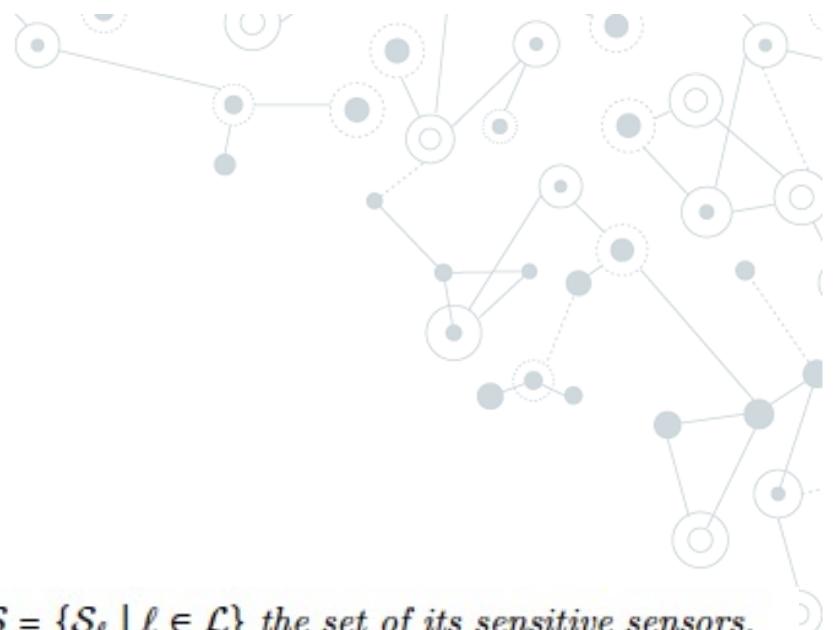


CFA: an untrustworthy node

Definition 2.3. Let N be a system of nodes with labels in \mathcal{L} , and $\mathcal{T} = \{\mathcal{T}_\ell \mid \ell \in \mathcal{L}\}$ be the set of tamperable sources. Then, the variable x of a node $\ell \in \mathcal{L}$ is untrustworthy w.r.t. \mathcal{T} , if for all derivatives N' s.t. $N \rightarrow^* N'$ it holds that $\Sigma_\ell^h(x) \downarrow_2 \in \{\diamond, \oplus\}$ where Σ_ℓ^h is the store of ℓ in N' .

Theorem 2.4. Let N be a system of nodes with labels in \mathcal{L} , and let $\mathcal{T} = \{\mathcal{T}_\ell \mid \ell \in \mathcal{L}\}$ the set of tamperable sources. Then a variable x of the node ℓ is untrustworthy w.r.t. \mathcal{T} if $(\hat{\Sigma}, \kappa, \Theta) \models N$, and $\Sigma_\ell(x) \downarrow_2 \subseteq \{\diamond, \oplus\}$.

Abbiamo detto che quindi stiamo tracciando due cose, da una parte l'affidabilità dell'informazione che può essere alterata dalfatto che le informazioni sono di tipo tamperable e dall'altra invece la secrecy attraverso valori che sono sensitive e che potrebbero passare in chiaro così esponendone in maniera impropria il contenuto. Quindi concentriamoci sulla prima proprietà e quindi la prima proprietà sostanzialmente mi va a vedere quand'è che l'informazione che sto raccogliendo, e sono in qualsiasi punto del mio sistema, è affidabile. Allora la definizione è quella che mi dici quand'è che una variabile di un nodo si può considerare affidabile oppure no, quindi quello che stiamo facendo adesso è tradurre in termini formali quello che abbiamo già intuito, cioè che naturalmente una variabile non è affidabile quando è macchiata, cioè quando ha almeno una goccia di inaffidabilità quindi la definizione precisamente mi dice che se N è un sistema di nodi con etichette in \mathcal{L} e io queste etichette le ho classificate per cui so quali sono i nodi che hanno delle sorgenti di tipo tamperable allora una variabile non è affidabile quando per tutti i modi possibili in cui il sistema si evolve, quindi per tutte le possibili N' raggiungibili con numero star di volte da N , abbiamo che i valori astratti associati ad x , nella loro seconda componente, quella è la proiezione, la freccina con il 2 è la proiezione sul secondo valore della coppia del valore astratto, appartiene a questi due tag o direttamente tamperable oppure tamperable e anche sensitive, che questa è l'altra variabile e in questo caso questo mi dice che la variabile non è affidabile e questo lo sto vedendo attenzione sul lato dell' evoluzione del sistema. A questo punto io posso anche stabilire qual è la connessione tra l'analisi statica e al solito l'evoluzione dinamica, io quindi ho il teorema successivo che mi dice che se ho un sistema sempre di nodi con l'etichetta in \mathcal{L} e alcune di queste sono già state classificate come tamperable, una variabile è untrustworthy se la mia analisi mi dice che lo può essere, cioè io faccio l'analisi, quindi c'ho l'analisi sigma cappuccio kappa theta di N e la variabile x nella componente sigma cappuccio mi dà proprio la sua inclusione tra i tag che non sono quelli che vorrei.



CFA: no leaks

Definition 2.5. Let N be a system of nodes with labels in \mathcal{L} , and $\mathcal{S} = \{\mathcal{S}_\ell \mid \ell \in \mathcal{L}\}$ the set of its sensitive sensors.

Then N has no leaks w.r.t. \mathcal{S} if $N \rightarrow^* N'$ and, for all $\ell_1, \ell_2 \in \mathcal{L}$, there is no transition $N' \xrightarrow{\langle v_1, \dots, v_n \rangle}_{\ell_1, \ell_2} N''$ such that $v_{i_{\ell_2}} \in \{\diamond, \ddiamond\}$ for some i .

Theorem 2.6. Let N be a system of nodes with labels in \mathcal{L} , and $\mathcal{S} = \{\mathcal{S}_\ell \mid \ell \in \mathcal{L}\}$ the set of its sensitive sensors.

Then N has no leaks w.r.t. \mathcal{S} if $(\hat{\Sigma}, \kappa, \Theta) \models N$, and $\forall \ell_1, \ell_2 \in \mathcal{L}$ such that $(\ell_2, \langle \hat{v}_1, \dots, \hat{v}_r \rangle) \in \kappa(\ell_2)$ we have that

$\forall i. \hat{v}_{i\ell_2} \in \{\diamond, \ddiamond\}$.

Vediamo invece come vedere la parte di invece di secrecy quindi quand'è che un nodo non ha leak. Quello che può succedere è quindi che io di nuovo ho un sistema di nodi, questa volta i sensori di tipo sensitive e quindi posso dire, questa è la definizione che il nodo non ha leaks quando per qualsiasi tipo di evoluzione che ha, non c'è mai una transizione da l1 a l2 che spedisce una tupla v1...vn tale che all'interno di questa tupla ci sia almeno un valore che è taggato o sensitive o sensitive anche tamperable, quindi questa analogamente a prima, quindi questo tipo di definizione mi dice esattamente cosa vuol dire a livello dinamico essere insicuri perché possibilmente leaking. A questo punto il teorema che mi mette in correlazione con quanto l'analisi mi calcola che cosa mi dice? Che in un caso di questo genere posso dire che non non ci sono leaks quando l'analisi del mio sistema mi va a vedere che per ogni tupla che esce da l1 verso l2, questa tupla non contiene mai informazioni di tipo sensitive.

L'ingrediente in più che c'è in questa formalizzazione qui, come vi avevo anticipato, è quello di legare questo flusso di informazione che possibilmente arriva colorato con il discorso delle decisioni nei punti critici o comunque su punti critici in cui vorrei invece avere dell'informazione che è o protetta oppure comunque affidabile.

L'esempio di gianburrasca: lì c'era un esempio carino di taint analysis, è la storia di un ragazzino impossibile che alla fine viene spedito in collegio perché è ingestibile e però piccolo sindacalista nato, si rende conto e sospetta che la minestra che viene data un certo giorno della settimana sia in realtà il risultato della rigovernatura dei piatti dei giorni precedenti quindi che sia una cosa truce, messa in atto dai cattivissimi gestori del collegio e per scoprirla versa dell'anilina che è una cosa di colore rosso dentro i piatti e scopre quindi il giorno dopo che la minestra arriva tutta rossa.



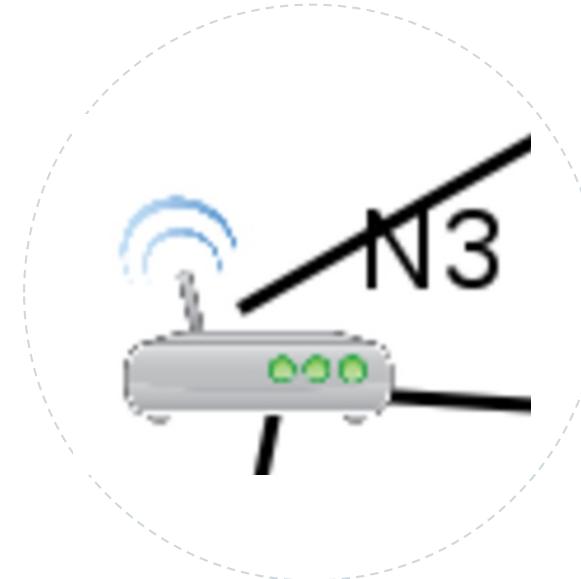
CFA: critical points

Definition 2.7. Let N be a system of nodes with labels in \mathcal{L} , $S = \{S_\ell \mid \ell \in \mathcal{L}\}$ the set of its sensitive sensors, $T = \{T_\ell \mid \ell \in \mathcal{L}\}$ the set of its tamperable sources, and \mathcal{P} a set of program critical points. Then N does not use tainted values in a critical point if $N \xrightarrow{*} N'$ and there is no transition $N' \xrightarrow{M_1^{a_1}, \dots, M_r^{a_r}}_\ell N''$ s.t. $a_i \in \mathcal{P}$ and $(\llbracket M_i^{a_i} \rrbracket_{\Sigma_\ell})_{\downarrow_2} \subseteq \{\diamond, \diamond, \diamond\}$ for some $i \in \{1, \dots, r\}$ and $\ell \in \mathcal{L}$.

Theorem 2.8. Let N be a system of nodes with labels in \mathcal{L} , $S = \{S_\ell \mid \ell \in \mathcal{L}\}$ the set of its sensitive sensors, $T = \{T_\ell \mid \ell \in \mathcal{L}\}$ the set of its tamperable sources, and \mathcal{P} a set of program critical points. Then N does not use tainted values in a program critical point if $(\hat{\Sigma}, \kappa, \Theta) \models N$, and $\Theta(\ell)(a)_{\downarrow_2} = \{\diamond\}$ for all labels $a \in \mathcal{P}$ and $\ell \in \mathcal{L}$.

Vediamo quindi come si gestisce la parte dei punti critici, quindi supponiamo che si stabiliscano dei punti critici e questo può essere fatto naturalmente anche dal designer del sistema, quindi è lui che sa quali sono le decisioni di attuazione più critiche, quali sono i punti in cui vorrei avere delle informazioni affidabile per fare i miei calcoli, quindi io suppongo di avere come al solito N S e T e avere anche un'insieme di punti critici nel programma. Posso dire che il sistema non usa valori di tipo taint nei punti critici individuati in un particolare punto critico, se per qualunque evoluzione di N non c'è mai una transizione che appunto manda in giro una tupla $M_1 \dots M_r$ tale che in uno dei punti critici, questi vengano utilizzati, tale che a_i appartenga ai punti critici e quindi in queste tuple non appare mai ovviamente il tag colorato appare solo quello nero, quindi questa è l'idea del critical point, quindi non c'è mai una transizione che implica il passaggio di informazione su più valori in cui uno di questi valori è considerato critico e l'informazione al tempo stesso è di tipo tagged e quindi è taint. A questo punto posso anche enucleare il teorema corrispondente, mettendo in relazione quello che voglio ottenere a tempo dinamico e cosa mi dice l'analisi dal punto di vista statico, quindi in analoga situazione io dico che N non usa valori di tipo taint in un punto critico del programma, quando è l'analisi stessa a dirmelo, perché analizza il tutto e fa sì che per ogni etichetta del punto critico, l'informazione che mi dà la componente theta in quel nodo, ovviamente quindi theta I di a quindi nella sua seconda componente mi restituisce solo roba untainted, questo mi dice sostanzialmente che possa star tranquillo, naturalmente la proprietà è sempre qui messa in negativo, quindi io sono sicura che se l'analisi in ogni punto critico mi dice che non ci sono valori taint io posso stare tranquillo, ovviamente se invece l'analisi mi dice che è possibile che in quel punto sostanzialmente arrivi, quindi si raggiunga quel punto con informazioni di tipo possibilmente taint, chiaramente mi suona il campanello d'allarme e vado a controllare davvero che cosa succede.

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

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

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

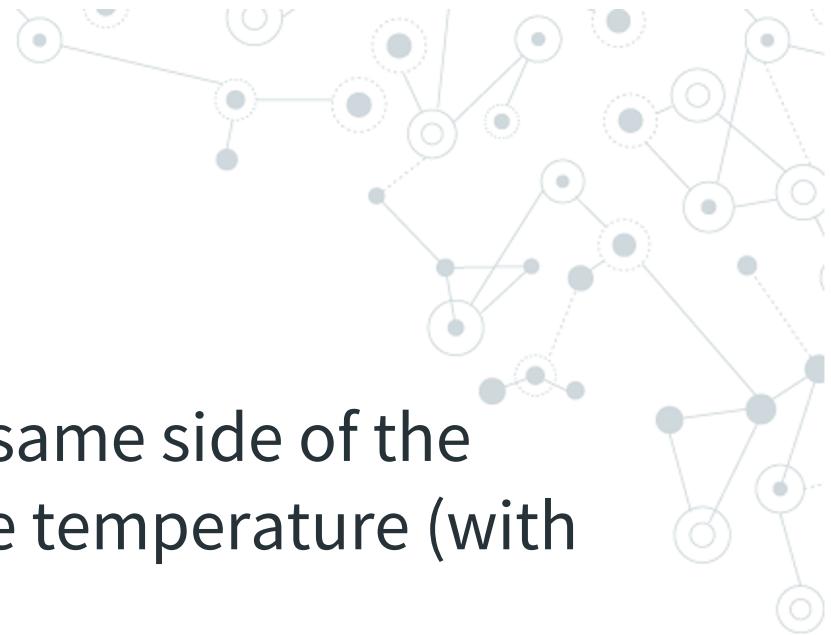


\lozenge is used to take decision

Questo è più o meno il tipo di ragionamento che quindi avevamo visto in termini intuitivi prima. Per cui io qui nell'esempio trovo che nel punto etichettato a posso dover ragionare su qualcosa che è potenzialmente macchiato e quindi mi devo fare i miei conti e decidere se conviene fare qualche controllo in più

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

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



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





Homework 1: resilience

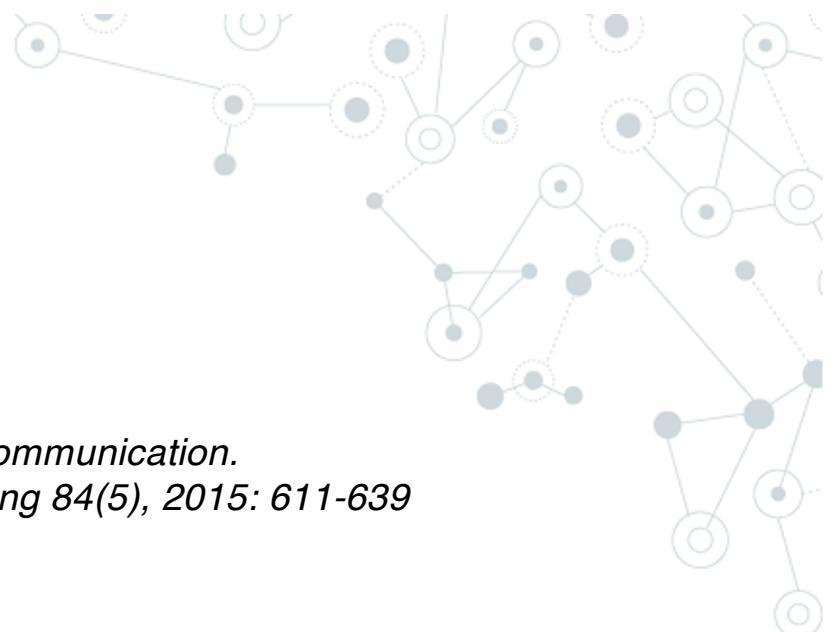
Resilience Analysis a là Quality The above taint analysis can be further pushed in order to consider resilience issues, by answering to question like “Given an aggregation function, how many values must be correct for giving the right answer?” The aim is to ensure a certain level of reliability of actuations even in the presence of unreliable data. For instance, consider the check whether the given temperature average, used to trigger the fire suppression system, is greater than a certain threshold, and suppose that the temperature sensors in the room now are eight.

$$gt(\text{avg}(1_{E_{11}}, 1_{E_{21}}, 1_{E_{31}}, 1_{E_{41}}, 1_{E_{51}}, 1_{E_{61}}, 1_{E_{71}}, 1_{E_{81}}), T_{RT})$$

A possible detailed question here may be: “What if $1_{E_{11}}$ and $1_{E_{61}}$, give the wrong results?” and “Does the overall result amortise these weaknesses?” Maybe it does, but “what does it happen if additionally, also $1_{E_{41}}$ fails?

Sviluppare analisi viste in due direzioni possibili, la prima ed ecco il motivo per cui abbia fatto vedere più da vicino l'analisi di tipo taint, è basato appunto sull'unione di questo tipo di informazioni ad un'altro tipo di informazione che ha a che fare con la qualità e che adesso vi faccio vedere e il secondo invece ha a che fare con l'analisi indipendentemente dal valore taint oppure no, ma delle traiettorie. Allora il primo concetto che vi volevo introdurre e che mi piacerebbe che combinaste con quella appena visto delle analisi di tipo taint è quello della resilienza. Quello che abbiamo visto finora, è un ragionamento sulle componenti taint o meno di un calcolo che possono avere conseguenze su situazioni critiche o su decisioni critiche. Che cosa succede quando parte delle informazioni può essere appunto inaffidabile? Allora la domanda giusta da farsi è: si cerco di proteggere il più possibile le mie fonti, però ci sono delle situazioni in cui è un po' difficile proteggere tutti i sensori da tutte le parti, dai possibili intrusioni, da possibili manomissioni di qualche attaccante. Allora l'idea è domandarsi invece quanto è affidabile, per esempio, il risultato di un'aggregazione in presenza di alcune cose che non sono affidabili, cioè quindi se io devo fare la media di una temperatura, quanto impatta, per esempio, il fatto che ci sia un sensore che potrebbe dare il risultato sbagliato, quindi questa è la domanda interessante perché mi dice quanto il mio sistema è resiliente rispetto a eventuali attacchi o anche se volete, anche se qui ormai ci stiamo muovendo rispetto alla sicurezza, ma se ci pensate anche rispetto al fault tolerance, cioè quanto regge anche un errore accidentale, perché immaginate che un sensore può dare il valore sbagliato perché manomesso, ma anche banalmente perché si è rotto qualche parte del meccanismo.

L'idea è di trovare un modo di combinare il fatto che da una parte io so già che alcune cose sono scippate o possibilmente scippate e dall'altra quant'è il livello di tolleranza del mio sistema a una certa percentuale di errore. L'idea è appunto quella di controllare questo tipo di informazioni, soprattutto in corrispondenza di attuazioni delicate. Quindi supponete di avere una stanza di un edificio con 8 sensori e che servono per decidere se c'è un incendio in corso eventualmente e quindi mettere in moto il sistema di antincendio. Quindi io calcolo la media di queste 8 temperature rilevate in quel momento e controllo se queste sono più grandi di una certa soglia, nel caso appunto questo sia vero, io devo mandare un allarme e applicare tutti i protocolli del caso. Quindi la domanda detta proprio brutalmente è: cosa succede se almeno due di questi mi danno il risultato sbagliato? Se questo è il caso, devo capire cosa può succedere, quindi come il risultato complessivo può ammortizzare questo risultato e se questo è vero, cosa succede se oltre a quei due se ne rompe un'altro? Quindi è un tipo di domande che ha a che fare con la qualità più che con il taint che mi dice quanto è resiliente, quanto tollera il mio sistema possibili errori, che è poi lo stesso problema che abbiamo enucleato sopra in termini intuitivi sui sensori del magazzino, cioè se io so che un sensore può essere manomesso, posso mettere in campo alcune strategie per minimizzare questo impatto e quindi posso dire davanti a queste strategie che anche se 1 su 4 mi da un risultato non affidabile, il mio tipo di attuazione comunque è al sicuro perché riesco a reggere questo livello di imprecisione.



Homework 1: binding operators

Vedi *H. Riis Nielson, F. Nielson, R. Vigo.*

A calculus of quality for robustness against unreliable communication.

Journal of Logical and Algebraic Methods in Programming 84(5), 2015: 611-639

$$\&_q(b_1, \dots, b_n)$$

Table 5
Quality predicates and their semantics.

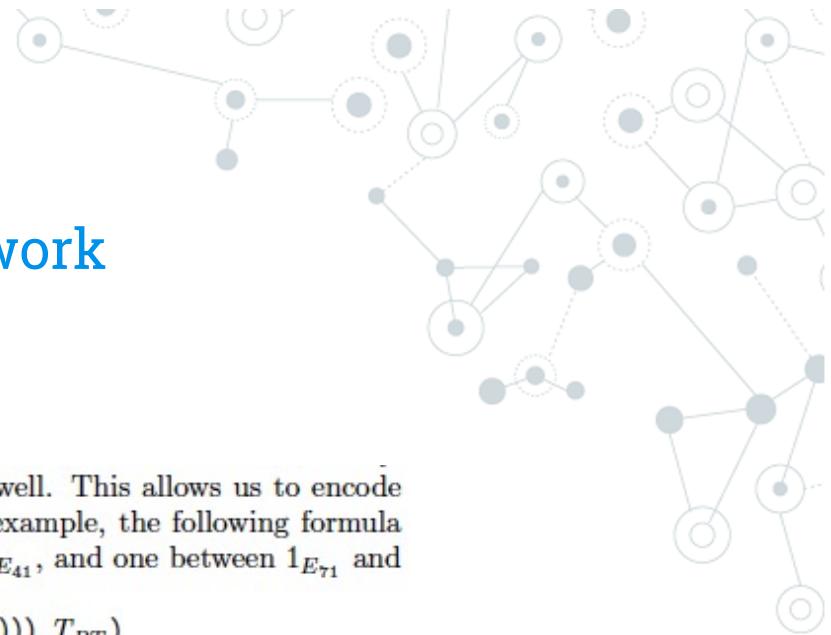
$[\{\forall\}](r_1, \dots, r_n) = (\{i \mid r_i = \text{tt}\} = n) = r_1 \wedge \dots \wedge r_n$
$[\{\exists\}](r_1, \dots, r_n) = (\{i \mid r_i = \text{tt}\} \geq 1) = r_1 \vee \dots \vee r_n$
$[\{\exists!\}](r_1, \dots, r_n) = (\{i \mid r_i = \text{tt}\} = 1)$
$[\{m/n\}](r_1, \dots, r_n) = (\{i \mid r_i = \text{tt}\} \geq m)$

$$\&_{\exists}(\hat{v}_1, \dots, \hat{v}_r) \text{ and } \&_{\forall}(\hat{v}_1, \dots, \hat{v}_r)$$

where $\&_{\exists}$ means that it suffices to have at least one correct value, and $\&_{\forall}$ means that all the values are necessary.

A questo punto come faccio tecnicamente? Vado a utilizzare un tipo di predici di qualità che sono stati introdotti da degli autori in Danimarca e che hanno in qualche maniera cablato dentro un calcolo che è diverso dal nostro, considerazioni di tipo qualitativo e quindi di tipo come abbiamo descritto, quindi quello che facciamo noi, e questa è una tipica mossa che si fa nell'ambito della ricerca, è contaminare quanto abbiamo fatto noi con ciò che è stato fatto dai loro per avere un'idea nuova, quindi l'idea è quella di prendere questi binding operators che usano loro e che sono sostanzialmente predici di tipo logico qualitativo (con cose tipo per ogni , esiste, non esiste eccetera eccetera) e li cabliamo all'interno del nostro framework e quindi ad esempio utilizziamo dei valori di tipo e-commerciale esiste v1...vr o e-commerciale per ogni v1...vr per dire che esiste almeno un elemento che ha una certa caratteristica oppure tutti gli elementi hanno una certa caratteristica con il significato legato proprio alla resilienza, come dicevamo quindi, in questo caso, l'elemento esistenziale mi dice che è sufficiente avere almeno un valore corretto oppure il for all significa che tutti i valori devono essere corretti perché la funzione di aggregazione o la tupla sia quella che serve. Naturalmente posso avere anche strutture logiche anche annidate con questi parametri che hanno comunque lo stesso tipo di valenza, quindi io, per esempio potrei avere questo tipo di struttura qui che mi dice che deve essere affidabile, per esempio, il valore di questi elementi sia di v1 che della condizione logica successiva, la condizione logica successiva mi dice che per essere vera almeno uno dei due v2 e v3 deve essere valido, quindi questo vuol dire che v1 e v2 oppure v1 e v3 sono combinazioni accettabili. Tornando ad esempio nel caso dell'edificio con 8 sensori, io potrei dire che la funzione mi regge possibili imprecisioni a patto che i dati affidabili siano E11, E21, E51, E61 e almeno uno tra E31 ed E41 ed uno tra E71 ed E81. Quindi questo è un modo per aggiungere un'ulteriore livello di analisi a quanto fatto, cioè io mi posso chiedere quant'è il possibile grado di resilienza di ciascun punto di decisione critica. Vi chiediamo con un certo grado di libertà, di combinare la l'informazione di tipo taint con le informazioni di tipo qualitativo per capire, dato una certo grado di informazioni taint quanto può reggere il vostro sistema, questo è un po l'idea, però è libera, nel senso che vi chiediamo di ideare un sistema in cui queste due informazioni vengono messe insieme e vengono rese disponibili per fare un'analisi più raffinata delle due separatamente.

Homework 1: resilience in our framework



More complex nested logical structures like $f(\&_{\forall}(\hat{v}_1, \&_{\exists}(\hat{v}_2, \hat{v}_3))$ are possible, as well. This allows us to encode the dependence of our actuation on a suitable combination of reliable data. For example, the following formula expresses that the reliable data are $1_{E_{11}}$, $1_{E_{21}}$, $1_{E_{51}}$, $1_{E_{61}}$, one between $1_{E_{31}}$ and $1_{E_{41}}$, and one between $1_{E_{71}}$ and $1_{E_{81}}$:

$$gt(avg(\&_{\forall}(1_{E_{11}}, 1_{E_{21}}, \&_{\exists}(1_{E_{31}}, 1_{E_{41}}), 1_{E_{51}}, 1_{E_{61}}, \&_{\exists}(1_{E_{71}}, 1_{E_{81}}))), T_{RT})$$

Note that this technique can be used starting from the estimates of our CFA as they are, because certain quality predicates can be applied directly to the resulting abstract values in order to reason about resilience issues.

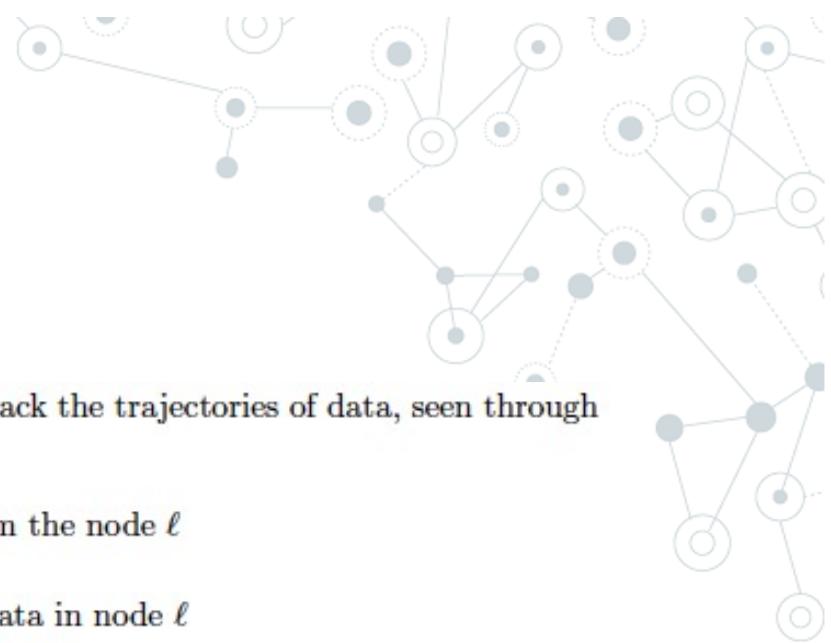
GOAL: Propose an extension of the framework in order to combine taint and quality information



Homework 2: trajectories

Try to reason about a way of extending the main CFA for IoT Lysa to track the trajectories of data, seen through the flows of abstract data in the CFA,

$\hat{v} ::=$	i^ℓ	data from sensor i from the node ℓ
	v^ℓ	constant in node ℓ
	$f^\ell(\hat{v}_1, \dots, \hat{v}_n)$	function on abstract data in node ℓ
	$\{\hat{v}_1, \dots, \hat{v}_n\}_{k_0}^\ell$	encryption on abstract data
	T^ℓ	term of depth greater than d

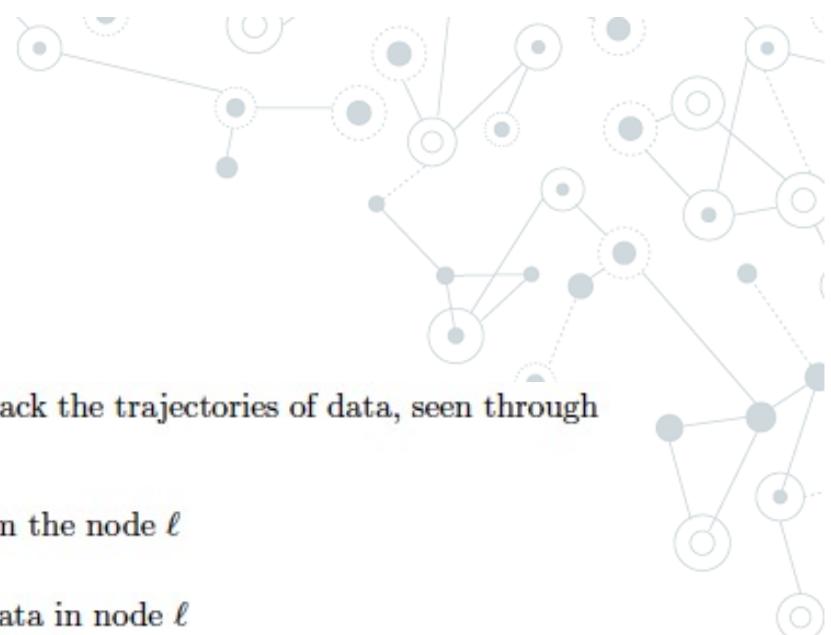


L'altra possibilità, invece riguarda le traiettorie, cioè noi abbiamo visto che si riesce un pochino a tracciare il percorso dei dati a livello astratto, in particolare voi sapete che una certa tupla può passare da un nodo ad un'altro, lo vedete dalla componente k , sostanzialmente se volete vederla in questo modo è come se aveste una micro traiettoria dal nodo cp al nodo a come vedete in questo esempio, del valore astratto $noiseRed$. Ora quello che chiediamo è: siccome avete tutte le micro traiettorie, sostanzialmente andando a vedere $k(l)$, quello che fosse possibile fare è provare a combinare le micro traiettorie, sempre con buona pace della precisione, in traiettorie, quindi se io vedo che $noiseRed$ passa da l_{cp} a l_a e poi l_a magari lo rimanda a l_s allora voi avete due traiettorie che possono essere combinate. Vi chiediamo di provare a pensare a come mettere insieme le micro traiettorie per formare proprio algebricamente, perché uno può pensare a un'insieme di micro traiettoria come si fanno a combinare in liste, quindi per cui da l_{cp} si va ad l_a , da l_a si va ad l_s etc etc.. quindi quello che vi chiediamo non è di fare un articolo nemmeno in questo caso, quindi è semplicemente di buttare giù delle ipotesi ragionevoli di trattamento di questo problema.

Homework 2: trajectories

Try to reason about a way of extending the main CFA for IoT Lysa to track the trajectories of data, seen through the flows of abstract data in the CFA,

$\hat{v} ::=$	i^ℓ	data from sensor i from the node ℓ
	v^ℓ	constant in node ℓ
	$f^\ell(\hat{v}_1, \dots, \hat{v}_n)$	function on abstract data in node ℓ
	$\{\hat{v}_1, \dots, \hat{v}_n\}_{k_0}^\ell$	encryption on abstract data
	T^ℓ	term of depth greater than d



Trajectories can be obtained, starting from a set of micro-trajectories (describing single hops) and by suitably composing them in order. In turn possibly micro-trajectories may be derived from investigating the κ component. For instance, from $\kappa(\ell_a) \ni (\ell_{cp}, \langle\!\langle noiseRed^{\ell_{cp}}(1^{\ell_{cp}}) \rangle\!\rangle)$ we can derive the micro-trajectory from N_{cp} to N_a of the abstract data represented by $noiseRed^{\ell_{cp}}(1^{\ell_{cp}})$.

This extension can allow us to state new security properties, such as ensuring that some (aggregated) data pass through an untrusted node before contributing to take a critical decision. The notion of trajectories can be extended also by associating a score to each node with label, representing some quantitative and logical information. Trajectories can be therefore compared on the basis of their overall score.



GOAL: Propose an extension of the framework in order to model trajectories of data