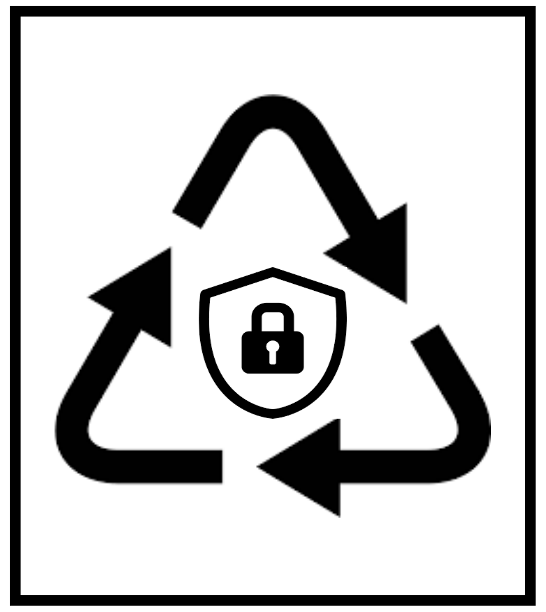
Towards Formalising Sustainable Security Systems

Liliana Pasquale

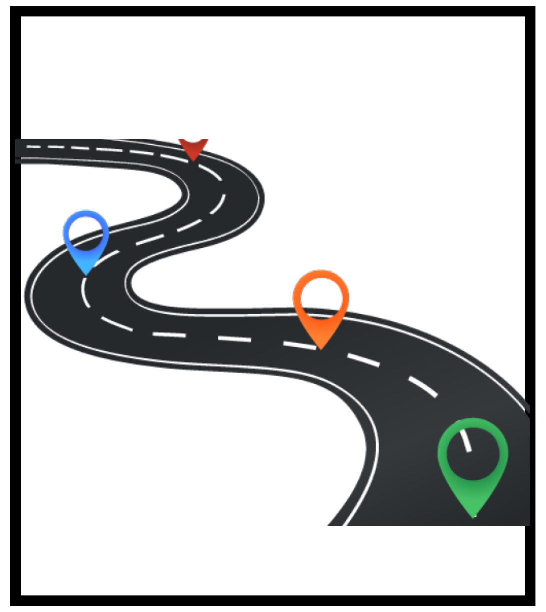
9th Conference on Formal Methods in Software Engineering (FormaliSE 2021)

Agenda



1 Introduction to Sustainable Security

2 3 Key Ideas to Support Sustainable Security



3 A Sustainable Security Roadmap

1 Introduction to Sustainable Security

Sustainability in Software Engineering

Software sustainability has been generally considered as the capacity of a software system to endure [Venters et al., 2017]

Sustainable Software:

[Beckers et al., 2015]

Software Engineering for Sustainability:

[Beckers et al., 2015]

Sustainability in Software Engineering

Software sustainability has been generally considered as the capacity of a software system to endure [Venters et al., 2017]

Sustainable Software: principles, practices and process that contribute to software endurance
[Beckers et al., 2015]

Software Engineering for Sustainability:
[Beckers et al., 2015]

Sustainability in Software Engineering

Software sustainability has been generally considered as the capacity of a software system to endure [Venters et al., 2017]

Sustainable Software: principles, practices and process that contribute to software endurance
[Beckers et al., 2015]

Software Engineering for Sustainability: Building software systems that support one or more dimensions of sustainability
[Beckers et al., 2015]

- Environmental
- Economical
- Individual
- Societal
- Technical.

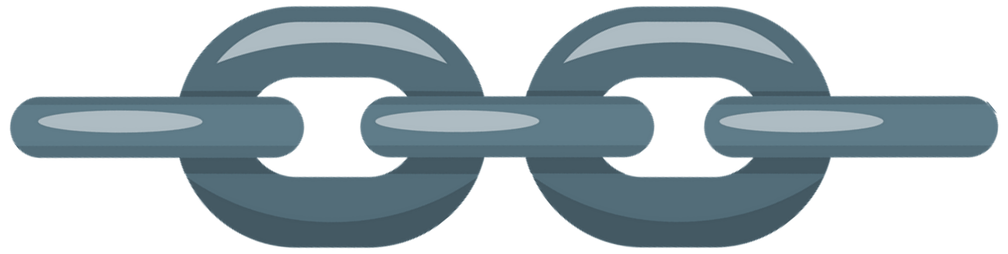
Sustainable Security

Software sustainability has been generally considered as the capacity of a software system to endure [Venters et al., 2017]

Sustainable Software: principles, practices and process that contribute to software endurance [Beckers et al., 2015]

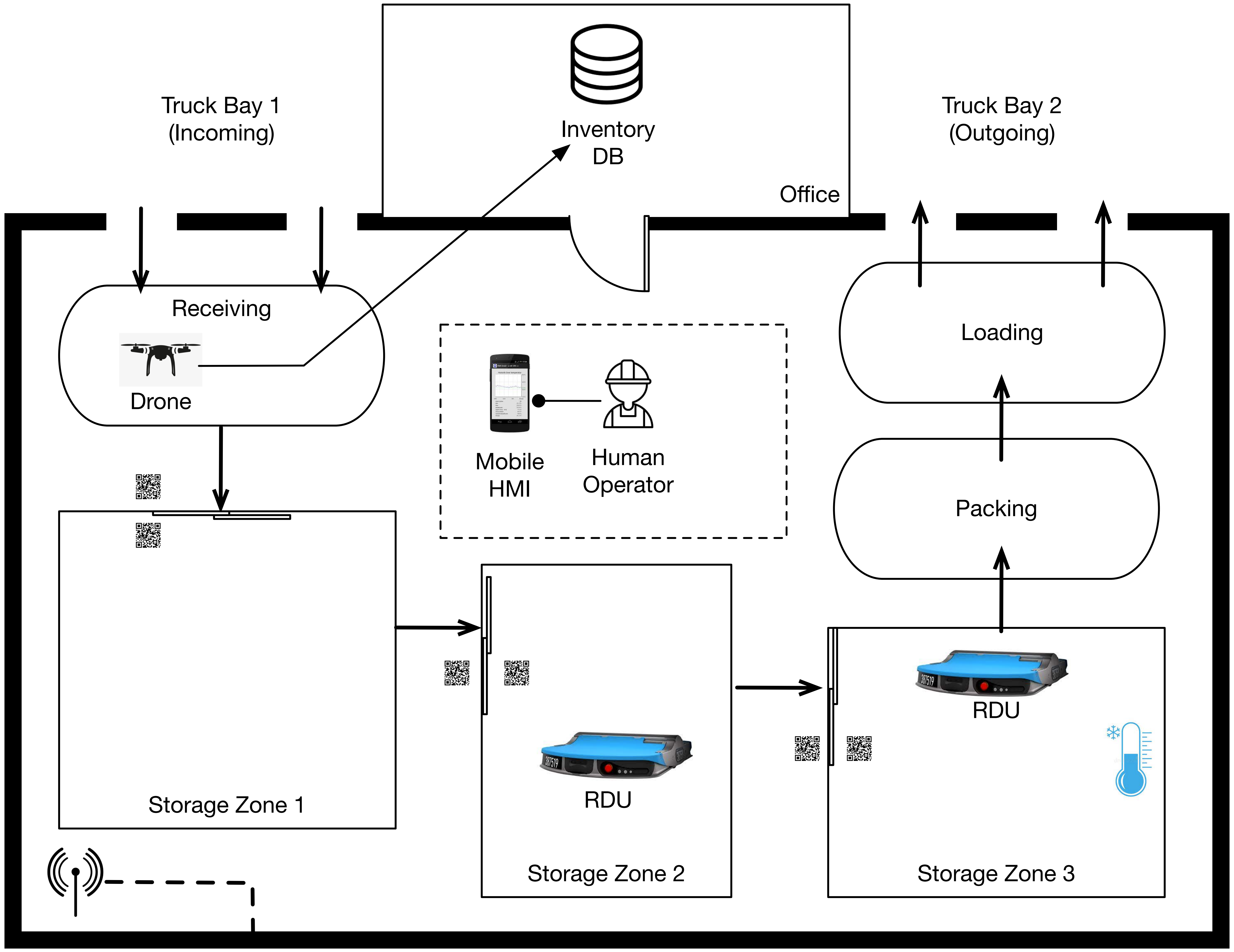
Sustainable Security

Capacity of Software to Endure Satisfaction of Security Requirements

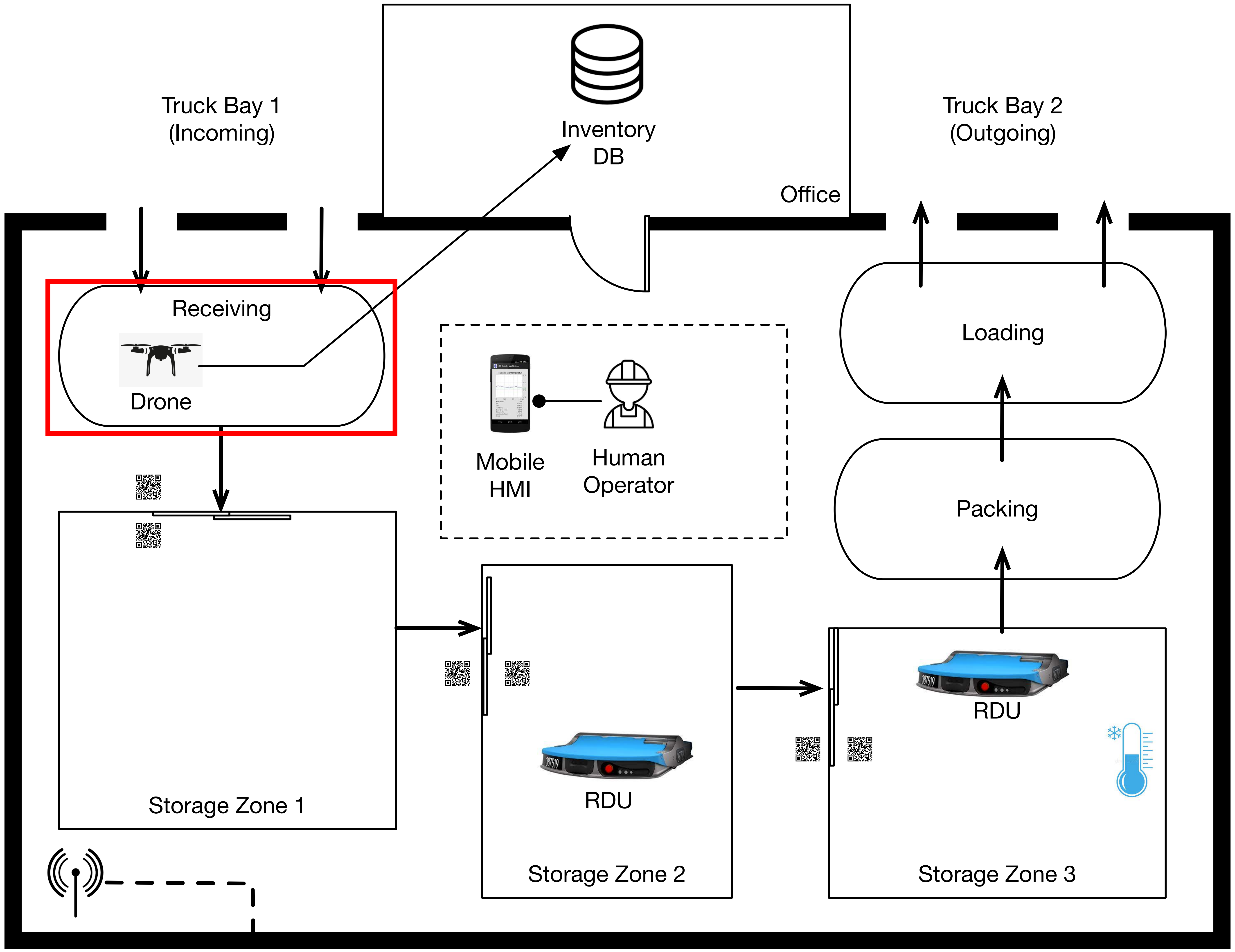
Security  Autonomy

Ensuring Sustainable Security in Cyber-Physical Systems is Challenging!

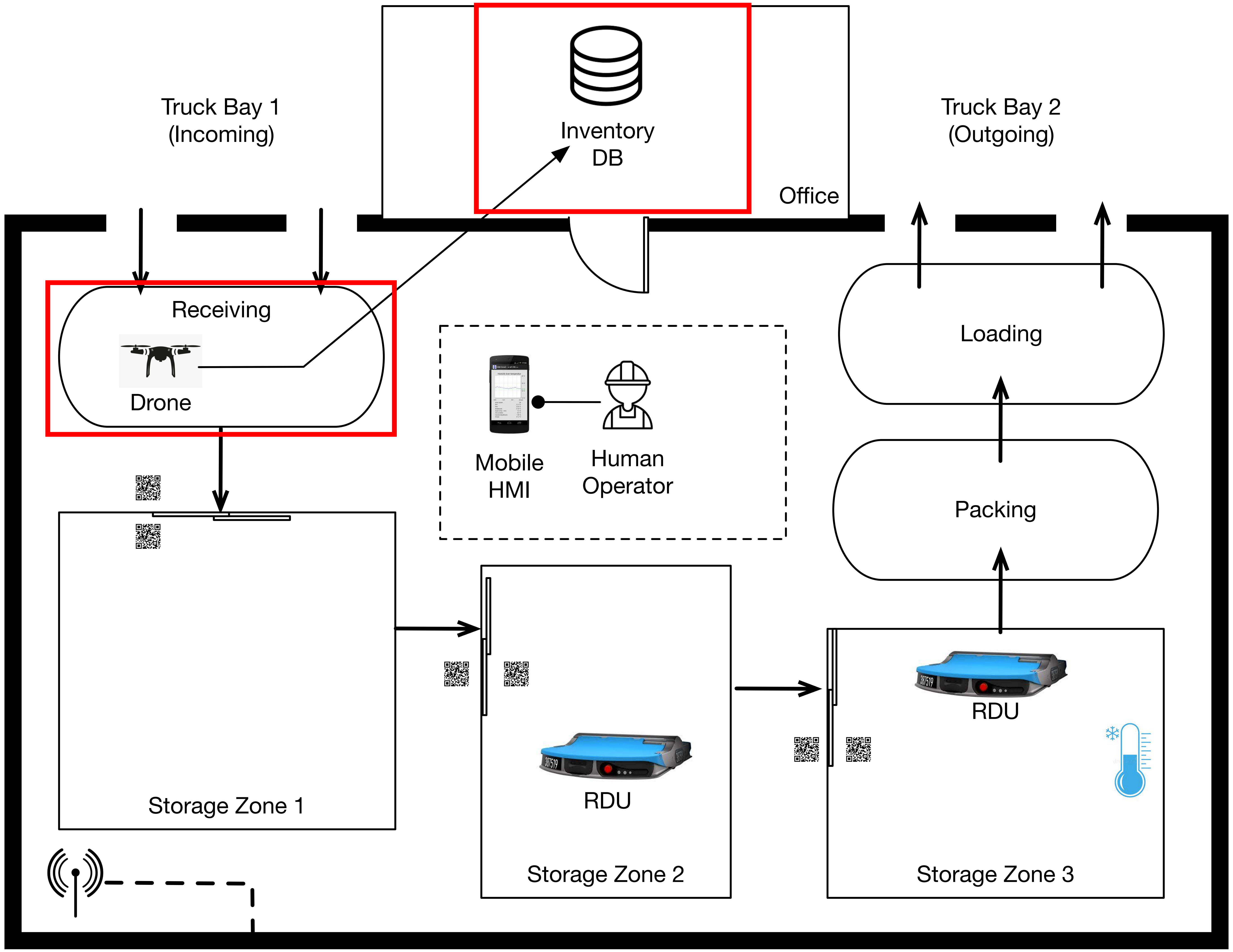
Warehouse of the zefirP Company



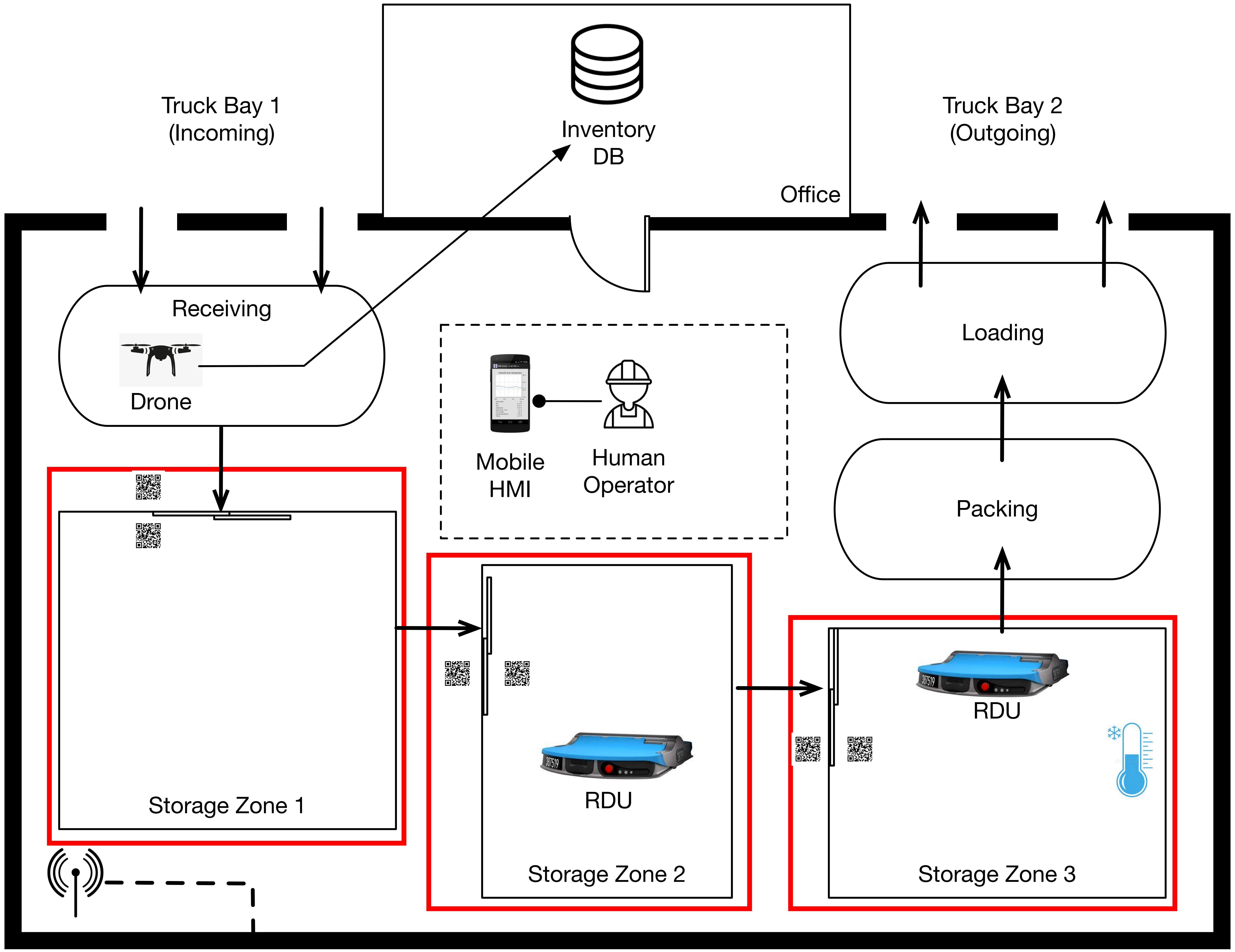
Warehouse of the zefirP Company



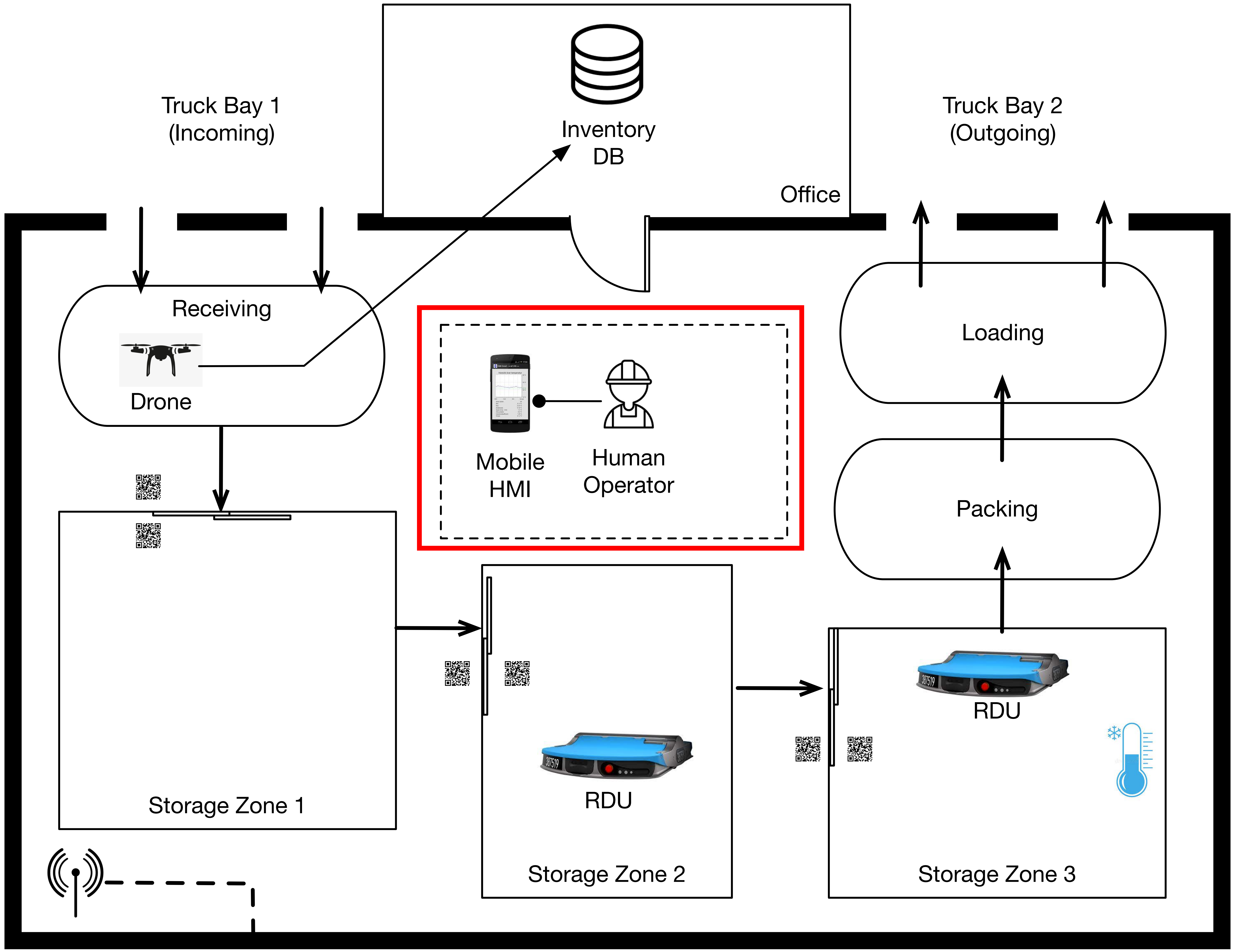
Warehouse of the zefirP Company



Warehouse of the zefirP Company

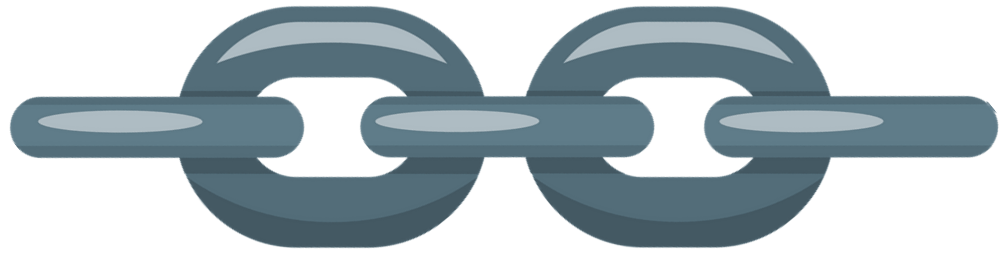


Warehouse of the zefirP Company



Sustainable Security

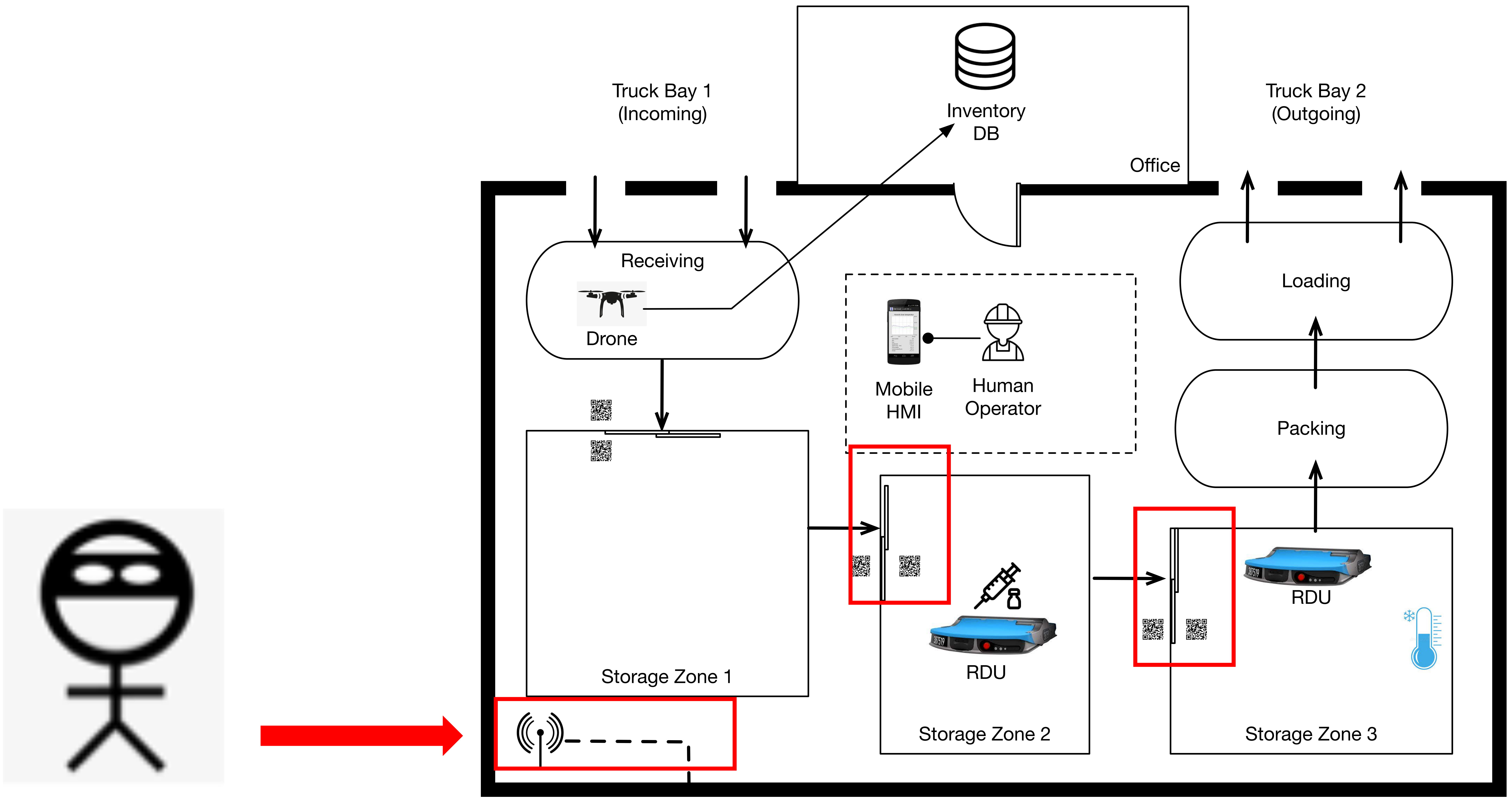
Capacity of Software to Continuously Satisfy Security Requirements

Security  **Autonomy**

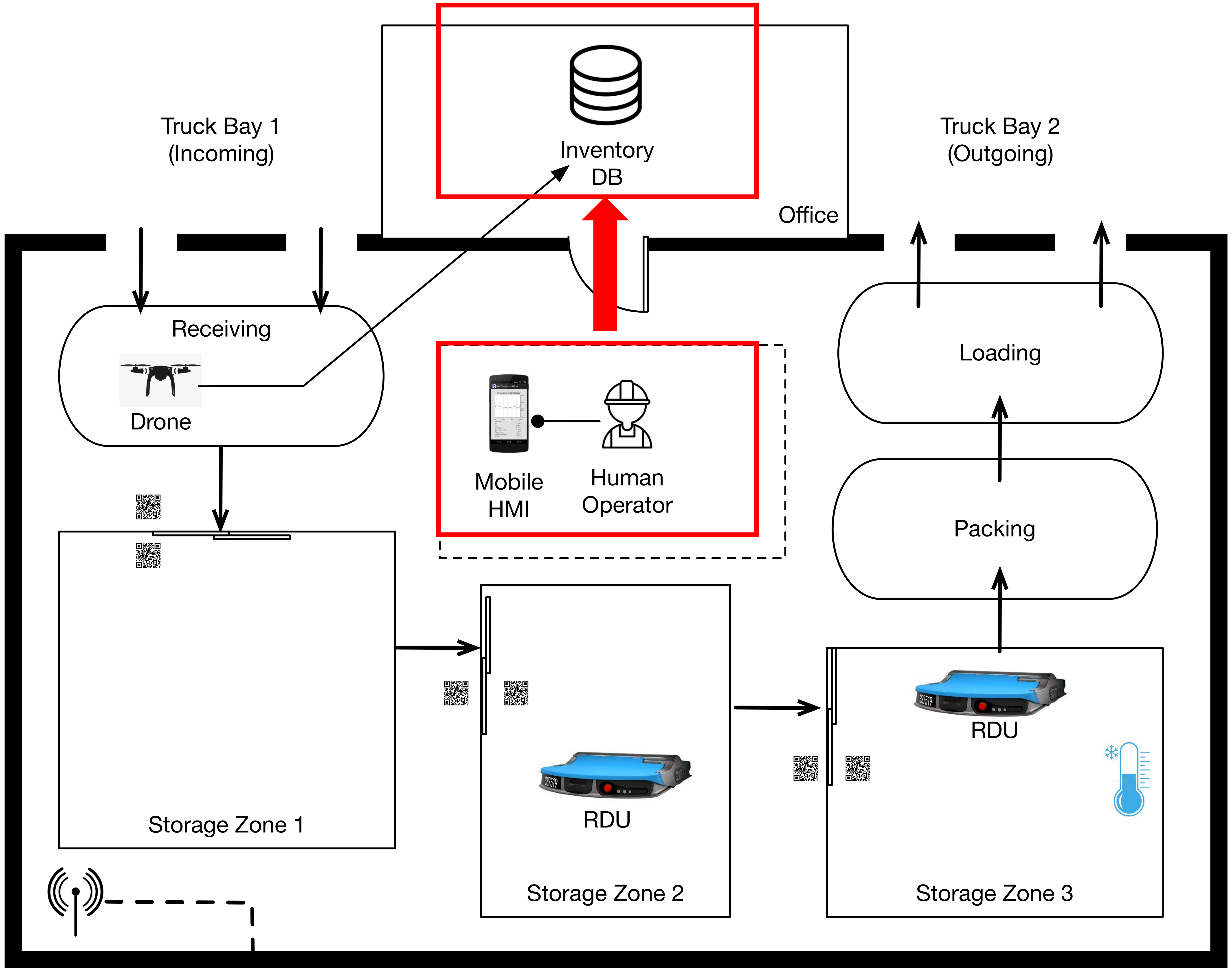
- Reason about and counteract threats brought by an **extended attack surface**

Extended Attack Surface: Cyber-Physical Threats

Extended Attack Surface: Cyber-Physical Threats

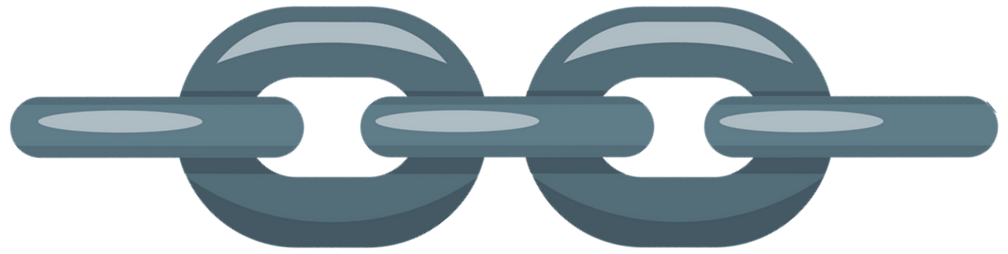


Extended Attack Surface: Physical-Cyber Threats



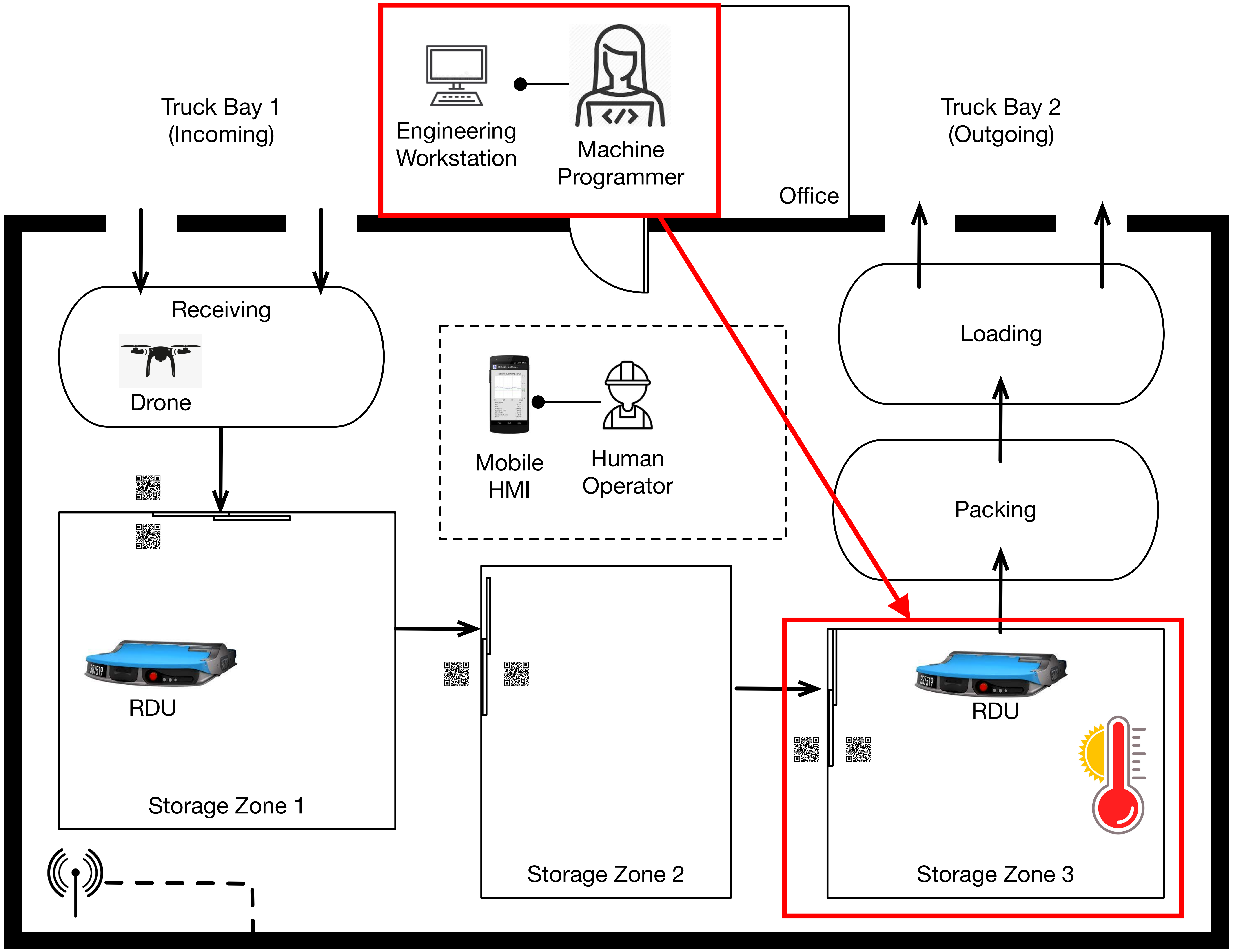
Sustainable Security

Capacity of Software to Continuously Satisfy Security Requirements

Security  **Autonomy**

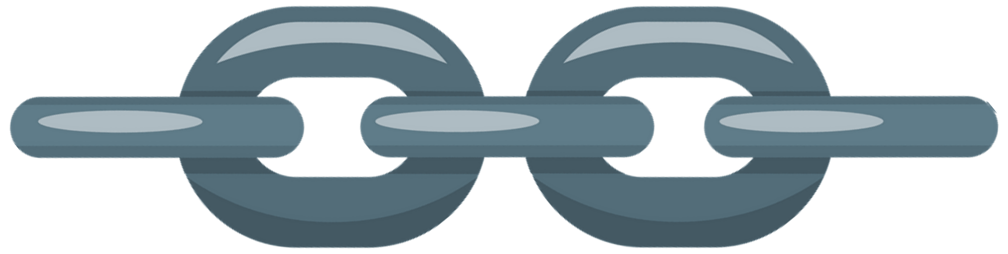
- Reason about and counteract threats brought by an **extended attack surface**
- Handle **uncertainties** brought by unexpected threats

Uncertainty: Unexpected Threats



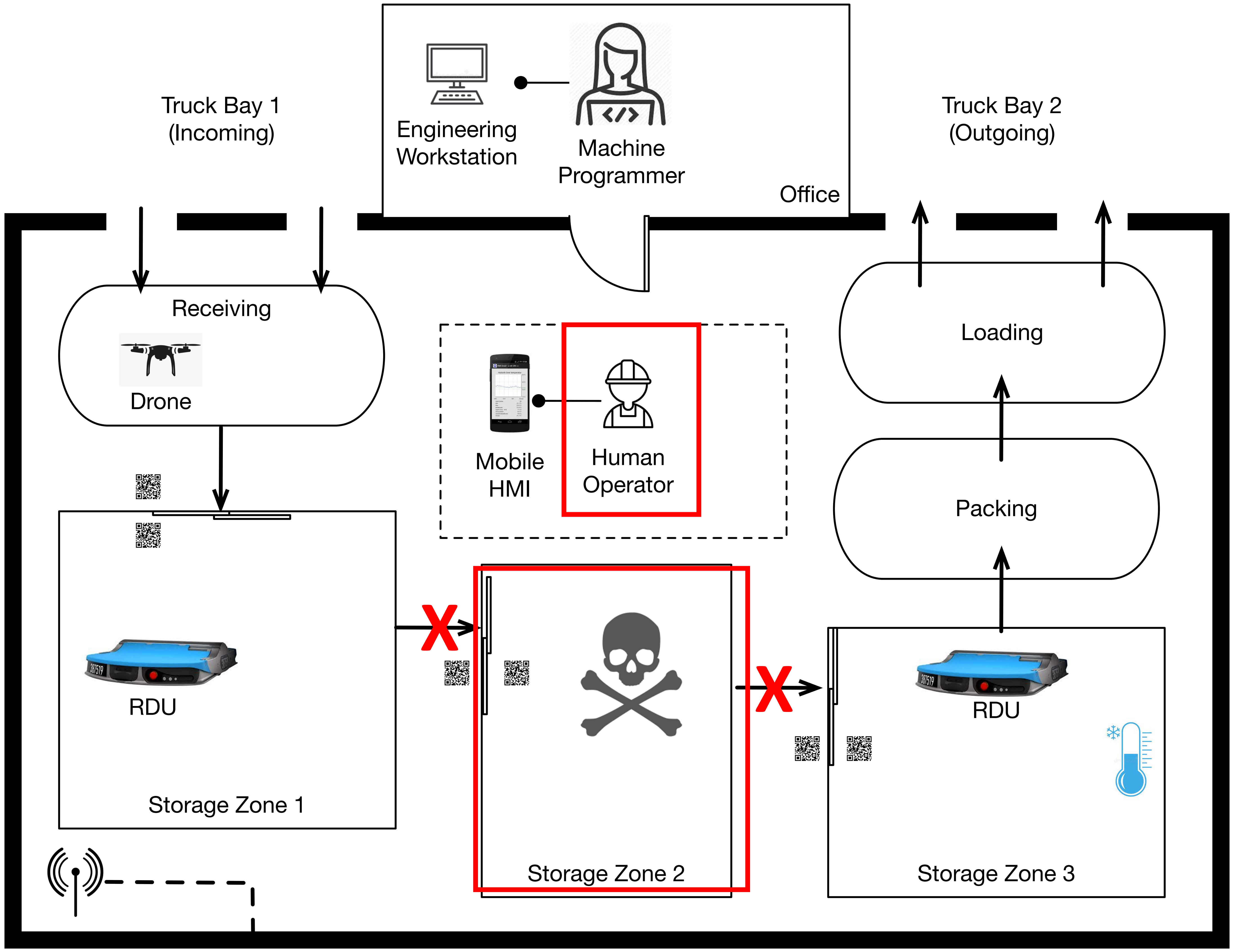
Sustainable Security

Capacity of Software to Continuously Satisfy Security Requirements

Security  **Autonomy**

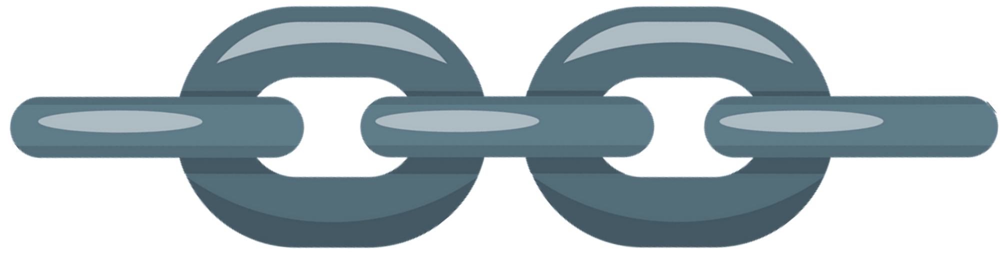
- Reason about and counteract threats brought by an **extended attack surface**
- Handle **uncertainties** brought by unexpected threats
- Endure **stakeholders' engagement**

Stakeholders: Human Operators need Explanations

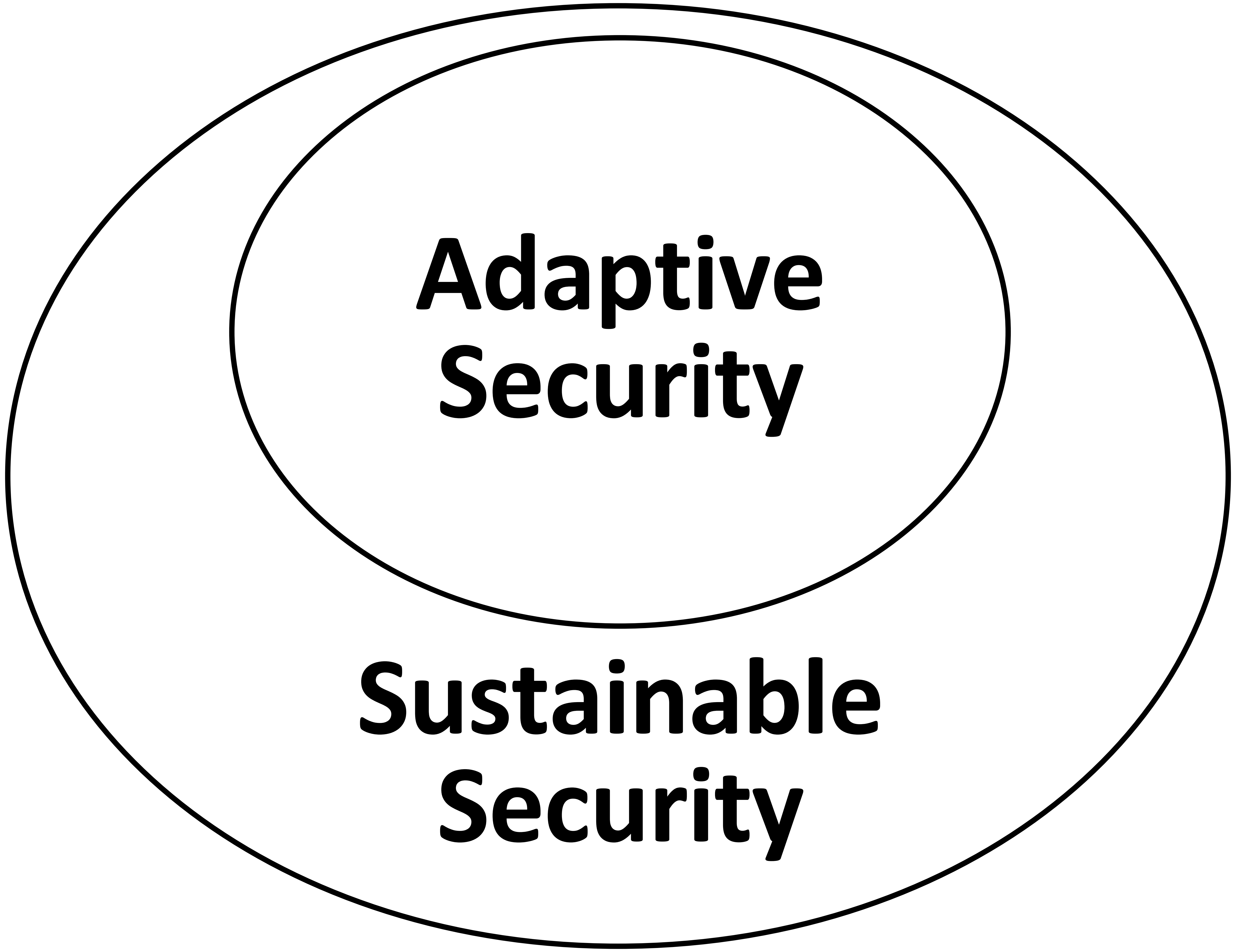


Sustainable Security

Capacity of Software to Continuously Satisfy Security Requirements

Security  **Autonomy**

- Reason about and counteract threats brought by an **extended attack surface**
- Handle **uncertainties** brought by unexpected threats
- Endure **stakeholders' engagement**


Adaptive
Security
Sustainable
Security

2 3 Key Ideas to Support Sustainable Security

- Formalize and reason about the extended attack surface
- Discover and counteract new threats
- Provide explanations to human operators

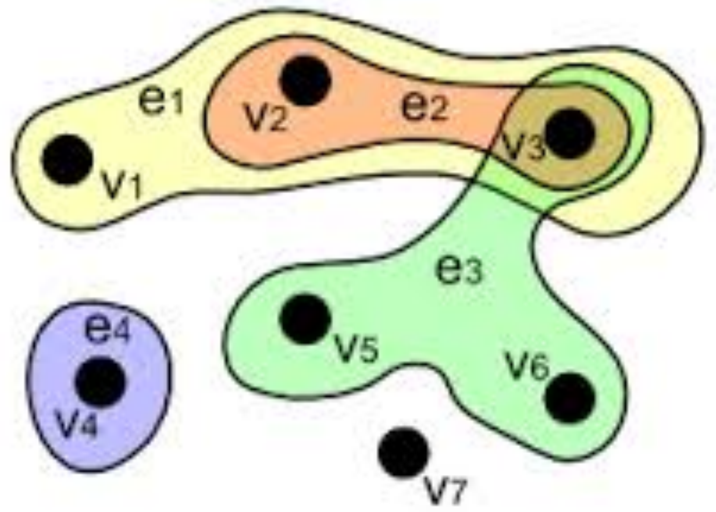
2

3 Key Ideas to Support Sustainable Security

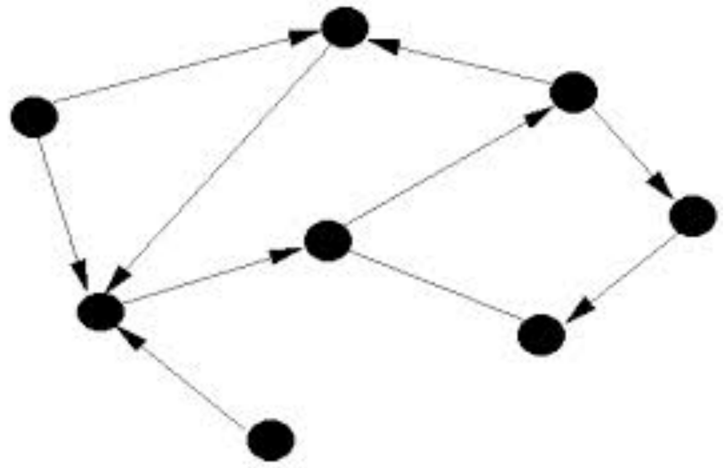
- **Formalize and reason about the extended attack surface**
- Discover and counteract new threats
- Provide explanations to human operators

Topology

Location
of objects and agents

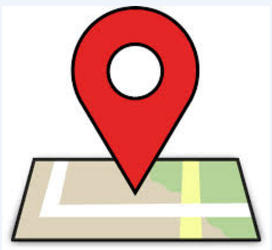


Structure of space

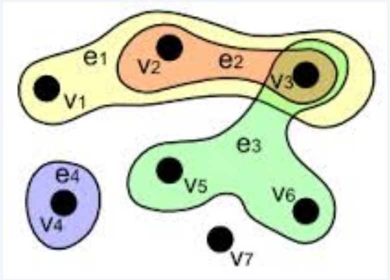


- **Proximity**
- **Reachability**

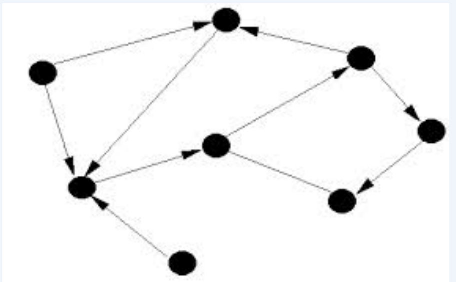
Topology



Location
of objects and agents



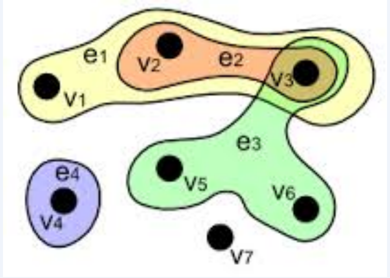
Structure of space



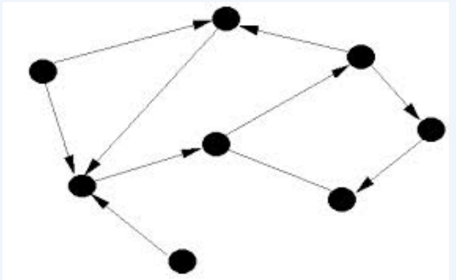
- **Proximity**
- **Reachability**

Topology

Location
of objects and agents

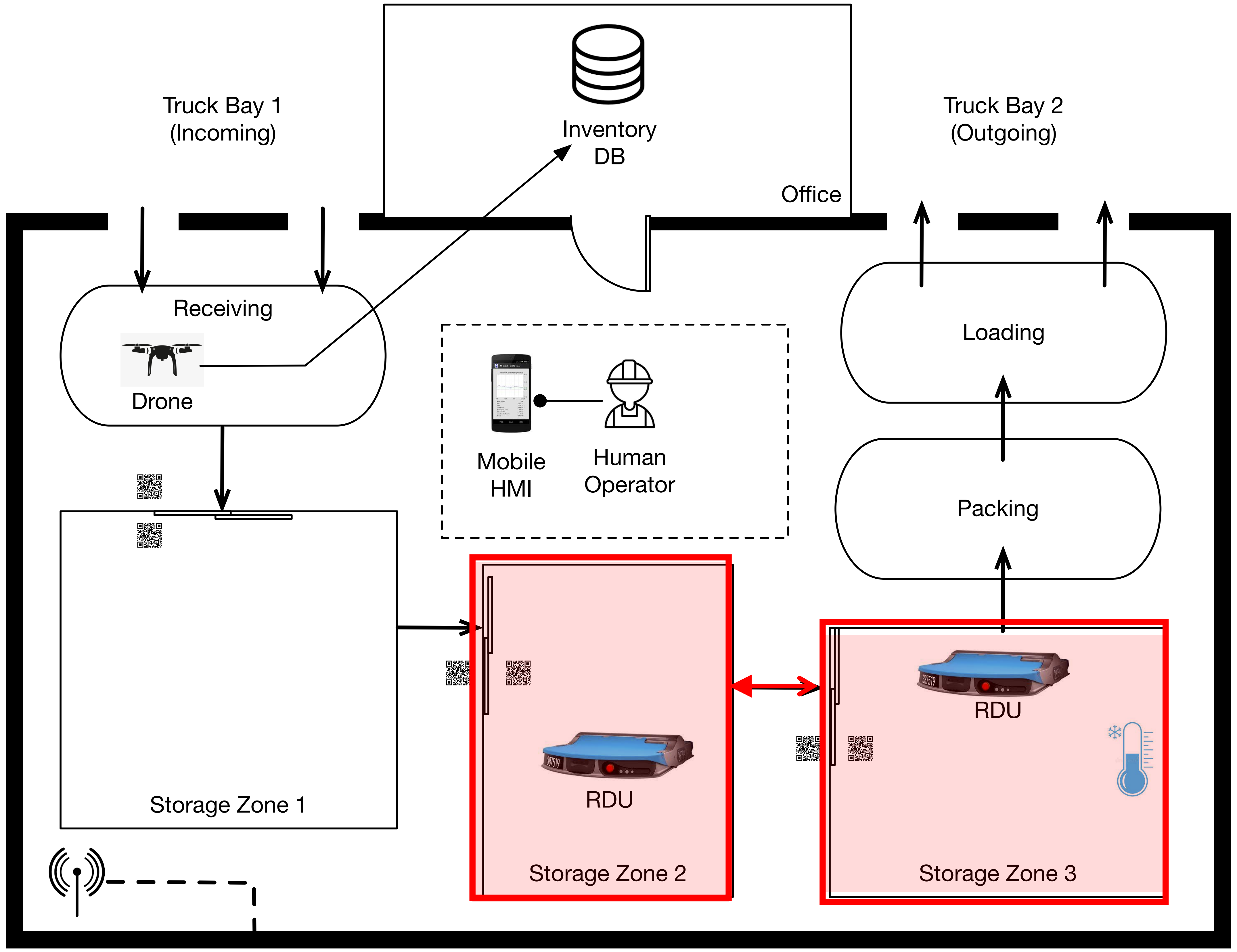


Structure of space

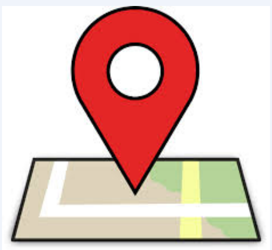


- **Proximity**
- **Reachability**

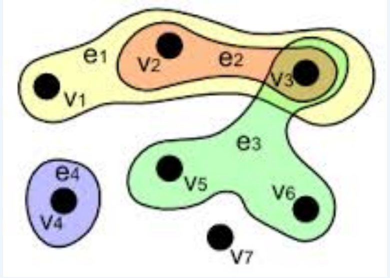
Connectivity
of physical areas.



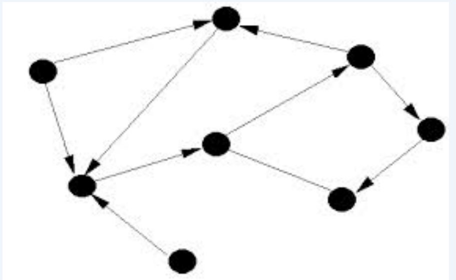
Topology



Location
of objects and agents

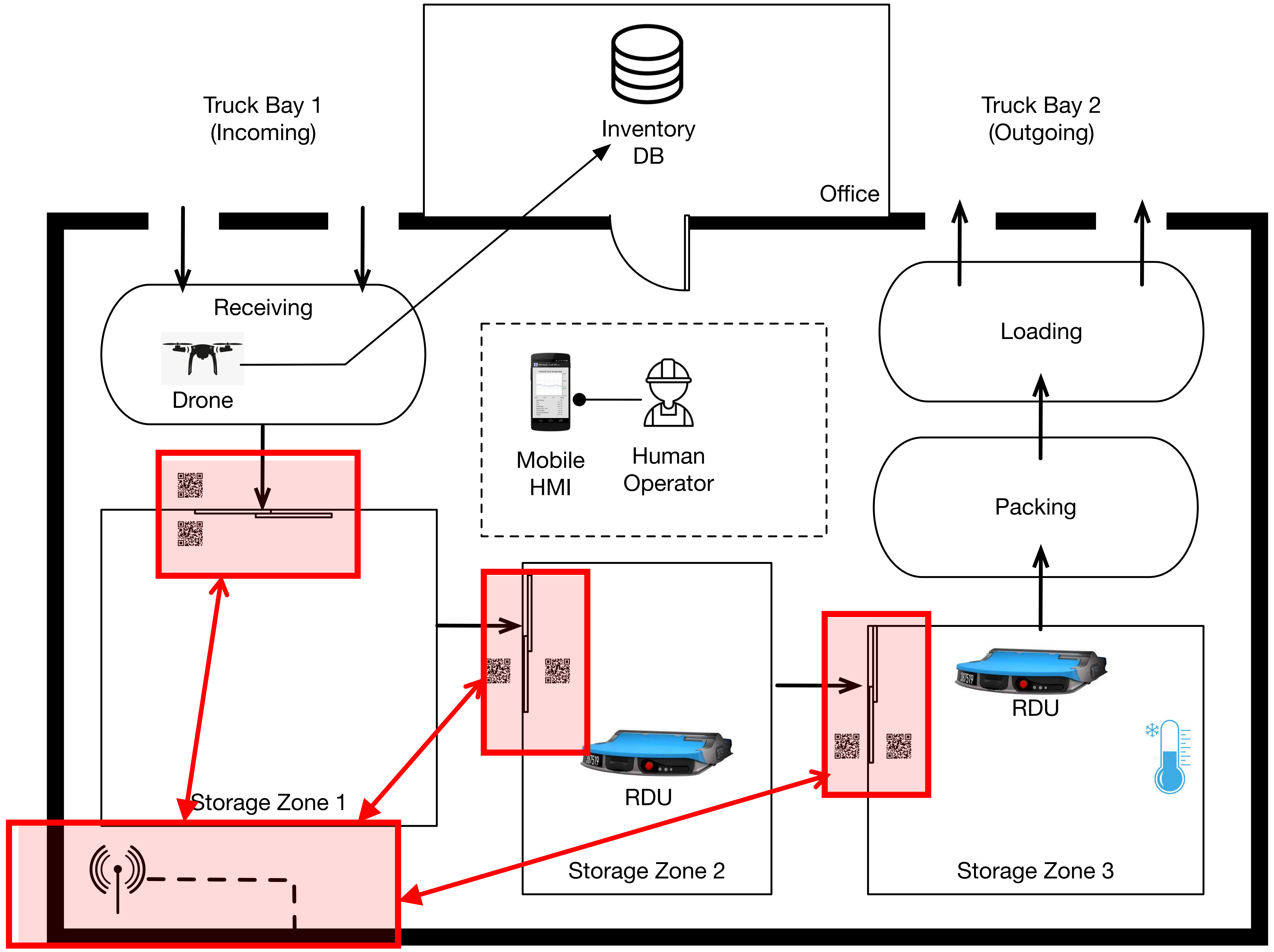


Structure of space



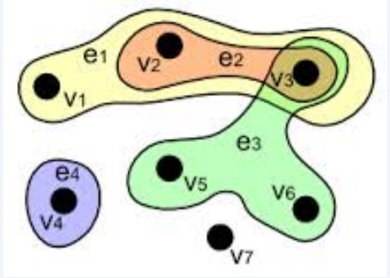
- **Proximity**
- **Reachability**

Connectivity
of digital areas.

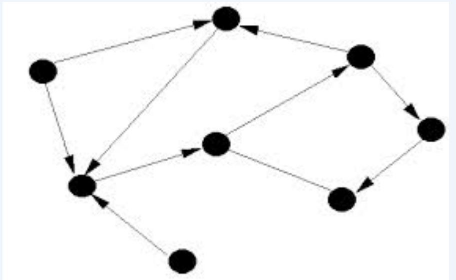


Topology

Location
of objects and agents



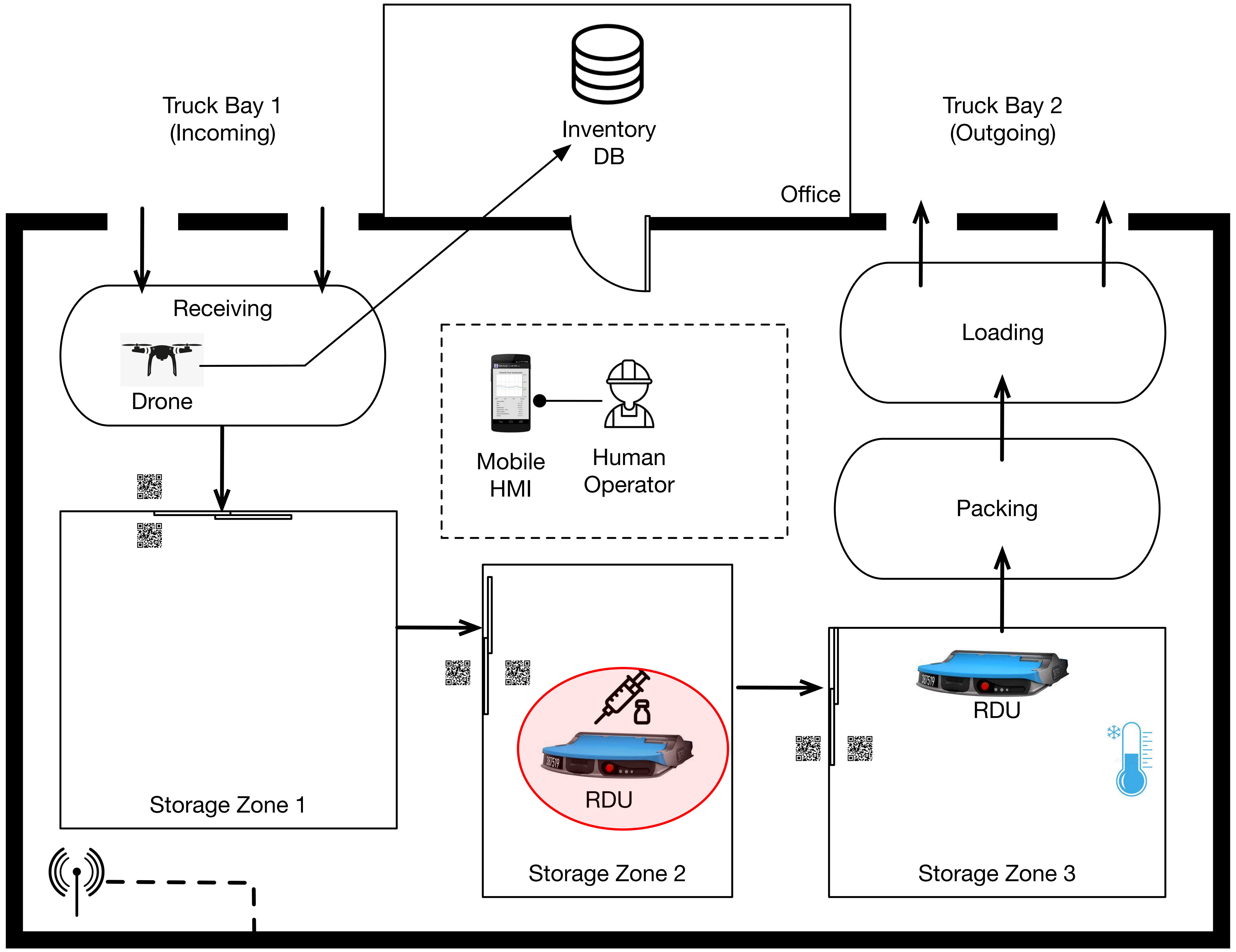
Structure of space



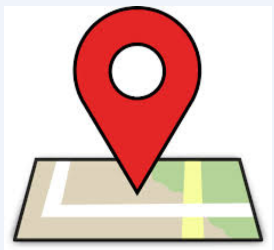
- **Proximity**
- **Reachability**

Placement
of objects and agents.

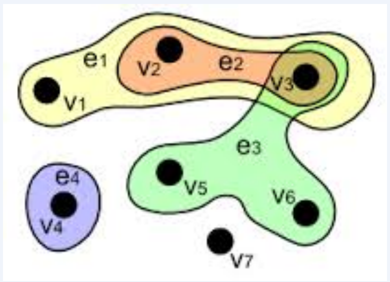
Connectivity
of physical/digital areas.



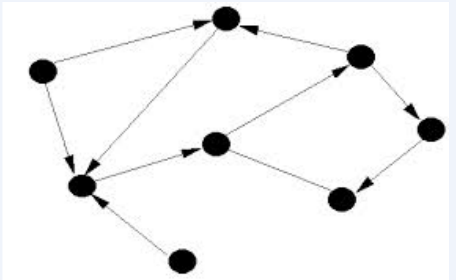
Topology



Location
of objects and agents



Structure of space

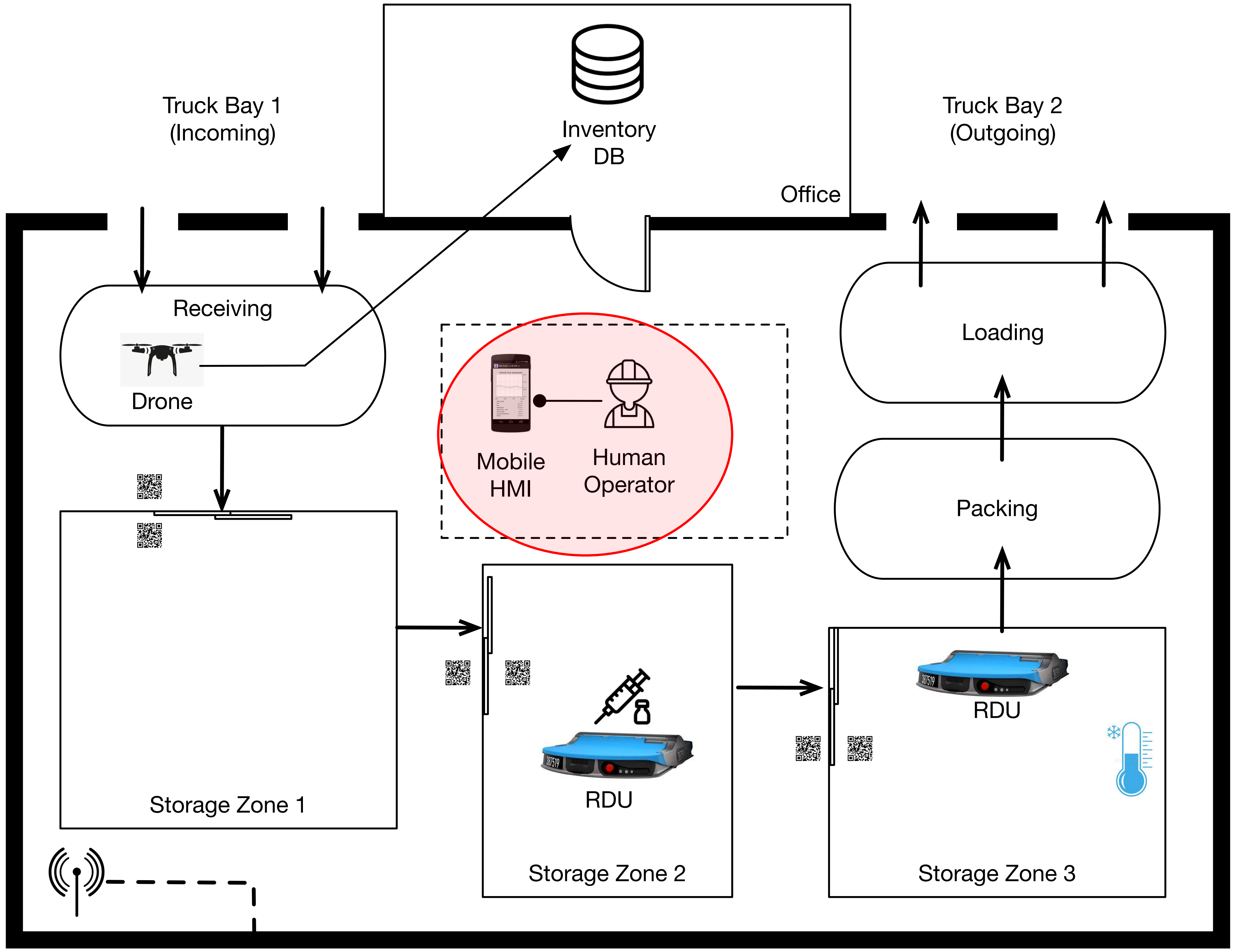


- **Proximity**
- **Reachability**

Placement
of objects and agents.

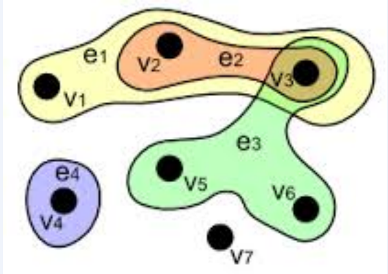
Connectivity
of physical/digital areas.

Proximity
Colocation in the same area.

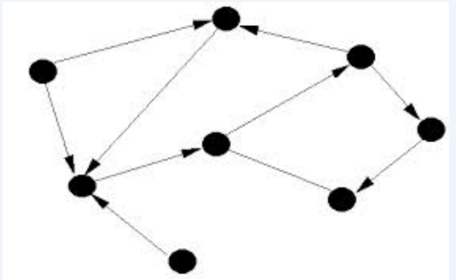


Topology

Location
of objects and agents



Structure of space



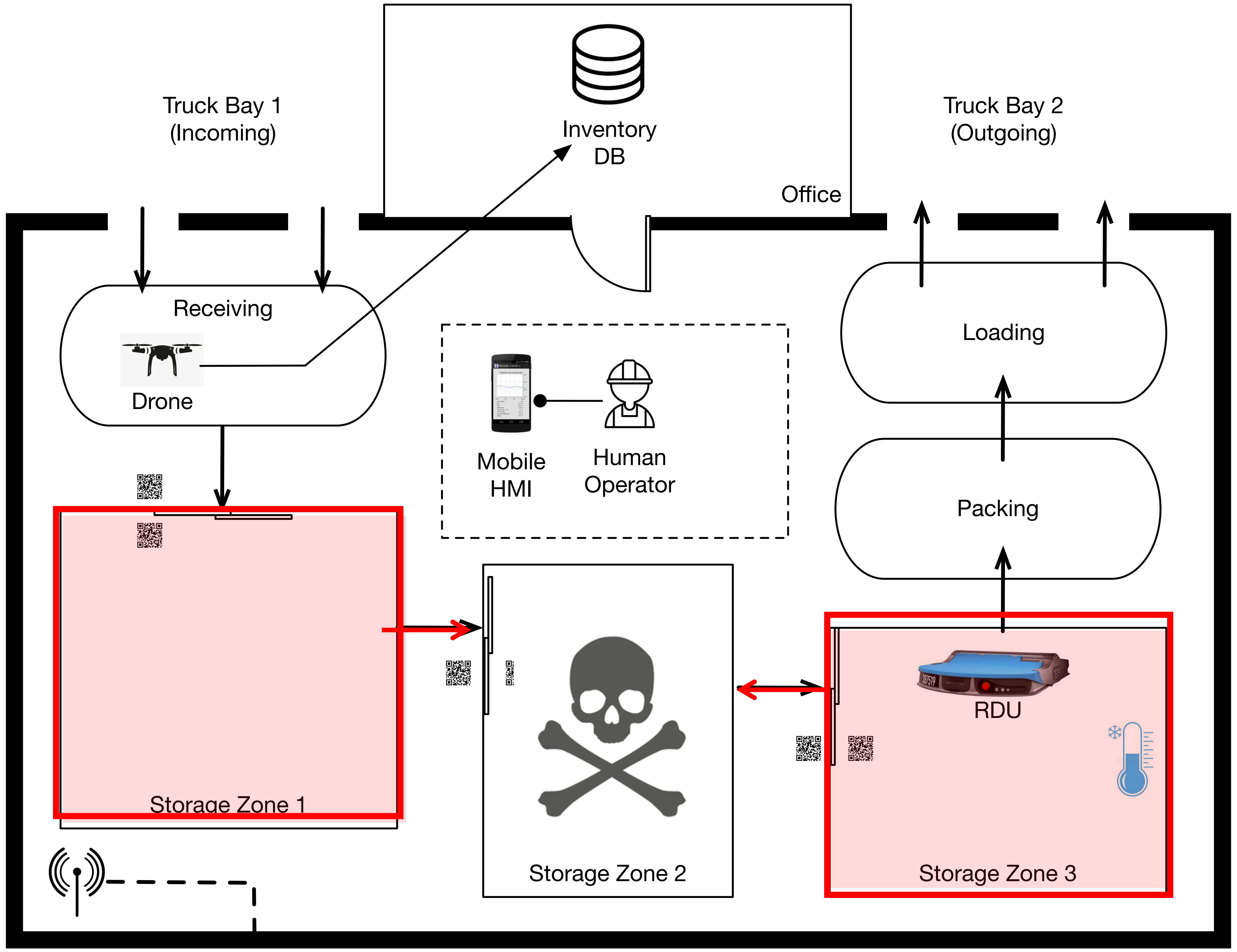
- **Proximity**
- **Reachability**

Placement
of physical objects and agents.

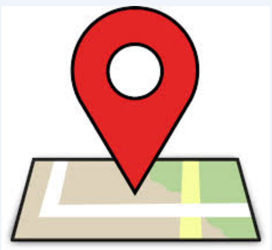
Connectivity
of physical/digital areas.

Proximity
Colocation in the same area.

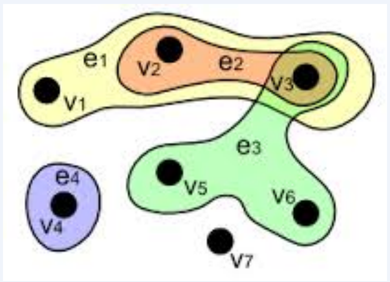
Reachability
Accessibility to physical areas or objects.



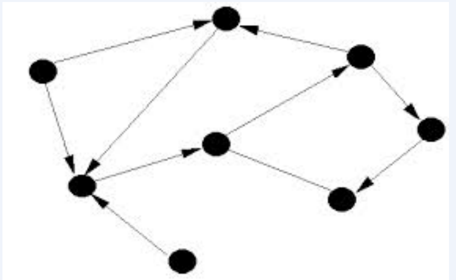
Topology



Location
of objects and agents



Structure of space



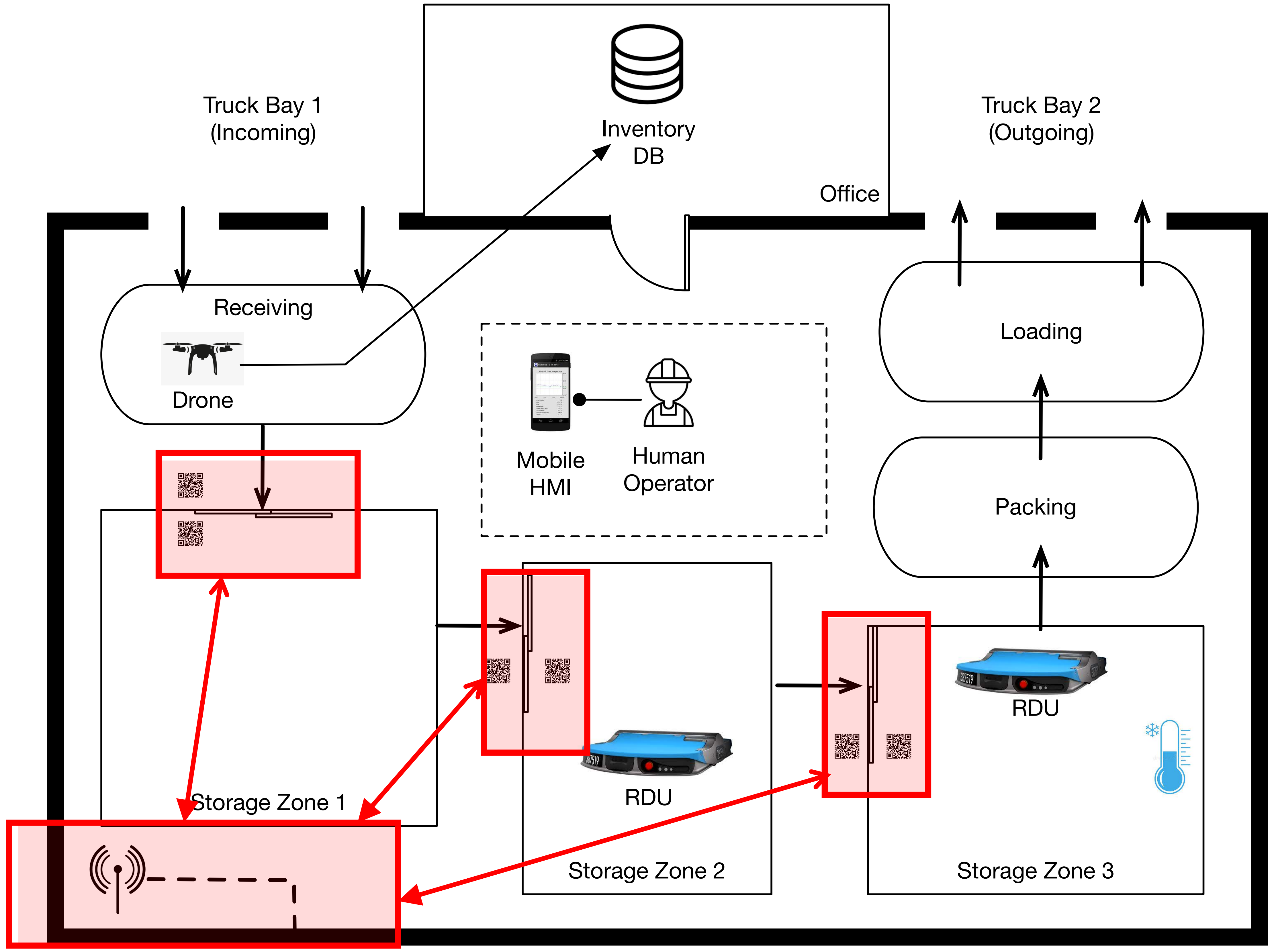
- **Proximity**
- **Reachability**

Placement
of physical objects and agents.

Connectivity
of physical/digital areas.

Proximity
Colocation in the same area.

Reachability
Accessibility to digital areas or objects.



Requirements for Modelling Topology

- Represent structure and communication
- Enable reasoning about the effects of topological changes

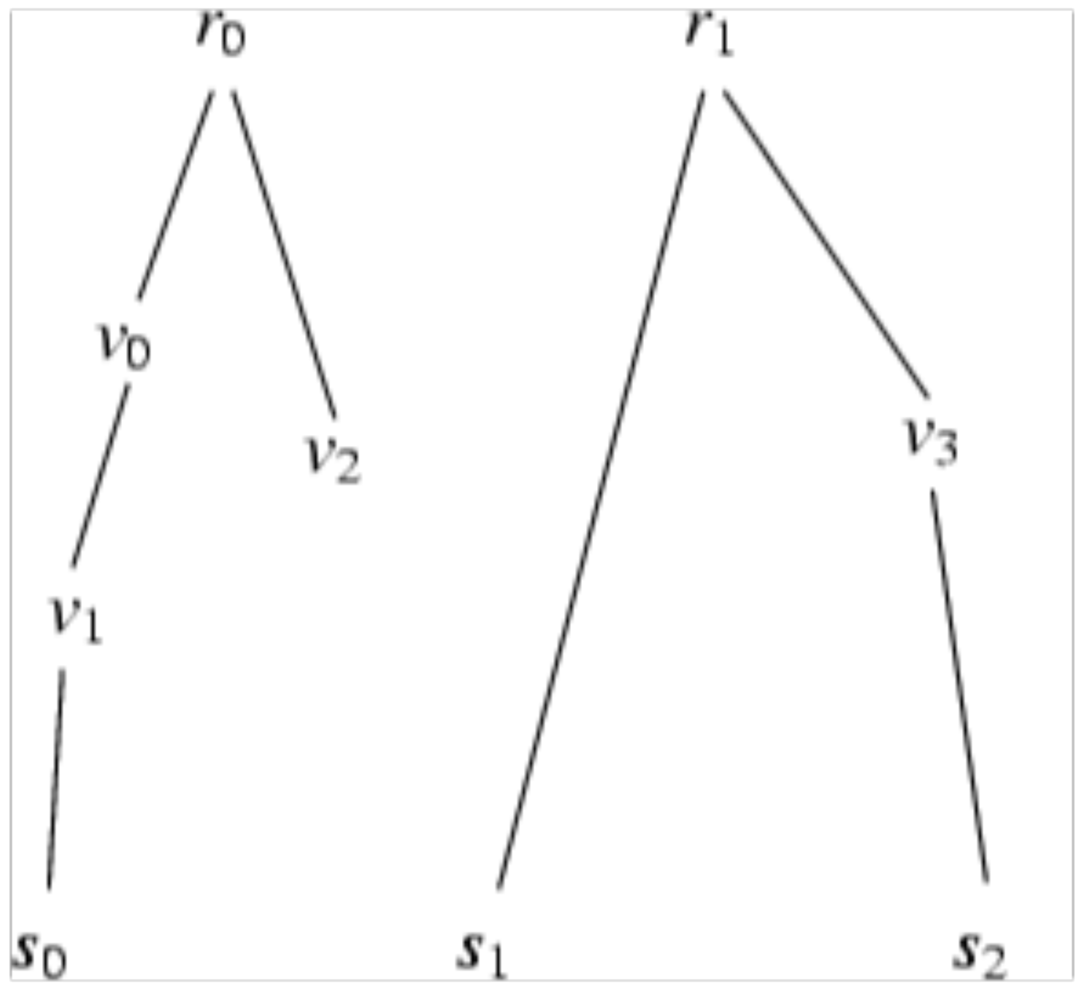
Process Calculi

- π-calculus
- Ambient Calculus [Cardelli & Gordon '98, Tsigkanos et al. '14]
- Bigraphical Reactive systems (BRS) [Milner '09] ☺
  - Extend bigraphs with well defined semantics of dynamic behaviour.

Bigraphs

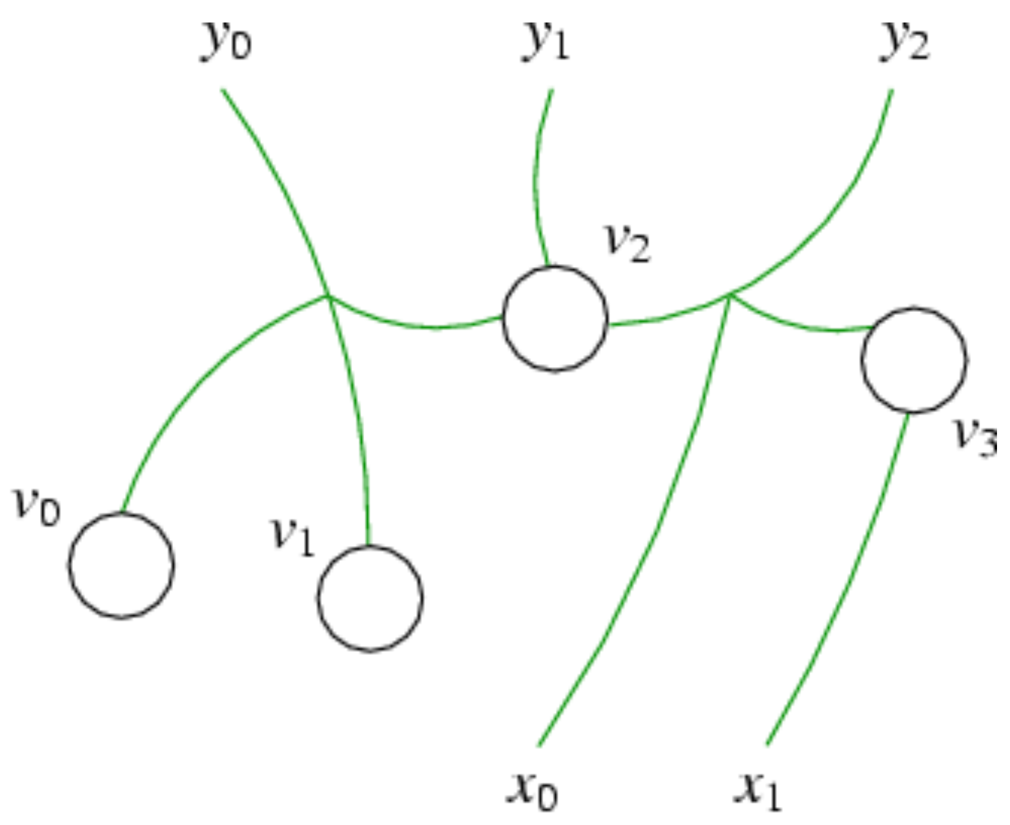
Place graph

- A forest of trees
- Nesting



Link graph

- A hypergraph of named edges over the set of nodes of the place graph
- Many-to-many relationships among nodes



Bigraphs – Algebraic Notation

$P.Q$	*Nesting (P contains Q)*
$-_i$	*Site numbered i*
$K_{\overrightarrow{w}}.(U)$	*Node associated with control K having ports with names in vector w. K contains U*
$/x.U$	*Ports with name x in U are connected*
$P \parallel Q$	*Juxtaposition of roots*
$P \circ Q$	*Composition*
$P \mid Q$	*Juxtaposition of children*

Bigraphs – Algebraic Notation

$P.Q$	*Nesting (P contains Q)*
$-_i$	*Site numbered i*
$K_{\vec{w}}.(U)$	*Node associated with control K having ports with names in vector w. K contains U*
$/x.U$	*Ports with name x in U are connected*
$P \parallel Q$	*Juxtaposition of roots*
$P \circ Q$	*Composition*
$P \mid Q$	*Juxtaposition of children*

$$(Room_{Office}.(Agent_{Mallory} \mid HMI_{wlink})) \parallel (Room_{wifiarea}.(Wifi_{wlink,lan} \mid (-_1)))$$
$$\parallel ((Room_{StgZone2,wlink}.(RDU_{1,wlink}.(Vaccine))) \mid Room_{StgZone3}.(Cooled))$$

Bigraphs – Algebraic Notation

$P.Q$	*Nesting (P contains Q)*
$-_i$	*Site numbered i*
$K_{\vec{w}}.(U)$	*Node associated with control K having ports with names in vector w. K contains U*
$/x.U$	*Ports with name x in U are connected*
$P \parallel Q$	*Juxtaposition of roots*
$P \circ Q$	*Composition*
$\boxed{P \mid Q}$	$\boxed{\textit{Juxtaposition of children}}$

$$\boxed{(Room_{Office}.(Agent_{Mallory} \mid HMI_{wlink}))} \parallel (Room_{wifiarea}.(Wifi_{wlink,lan} \mid (-_1)))$$

$$\parallel ((Room_{StgZone2,wlink}.(RDU_{1,wlink}.(Vaccine))) \mid Room_{StgZone3}.(Cooled)$$

Bigraphs – Algebraic Notation

$P.Q$	*Nesting* (P *contains* Q)
$-_i$	*Site numbered* i
$K_{\vec{w}}.(U)$	*Node associated with control* K *having ports with names in vector* w. K *contains* U
$/x.U$	*Ports with name* x *in* U *are connected*
$P \parallel Q$	*Juxtaposition of roots*
$P \circ Q$	*Composition*
$P \mid Q$	*Juxtaposition of children*

$$(Room_{Office}.(Agent_{Mallory} \mid HMI_{wlink})) \parallel (Room_{wifiarea}.(Wifi_{wlink,lan} \mid (-_1)))$$

$$\parallel ((Room_{StgZone2,wlink}.(RDU_{1,wlink}.(Vaccine))) \mid Room_{StgZone3}.(Cooled)$$

Bigraphs – Algebraic Notation

$P.Q$	*Nesting* (P *contains* Q)
$-_i$	*Site numbered* i
$K_{\vec{w}}.(U)$	*Node associated with control* K *having ports with names in vector* w. K *contains* U
$/x.U$	*Ports with name* x *in* U *are connected*
$P \parallel Q$	*Juxtaposition of roots*
$P \circ Q$	*Composition*
$P \mid Q$	*Juxtaposition of children*

$$(Room_{Office}.(Agent_{Mallory} \mid HMI_{wlink})) \parallel (Room_{wifiarea}.(Wifi_{wlink,lan} \mid (\_1)))$$

$$\parallel ((Room_{StgZone2,wlink}.(RDU_{1,wlink}.(Vaccine))) \mid Room_{StgZone3}.(Cooled)$$

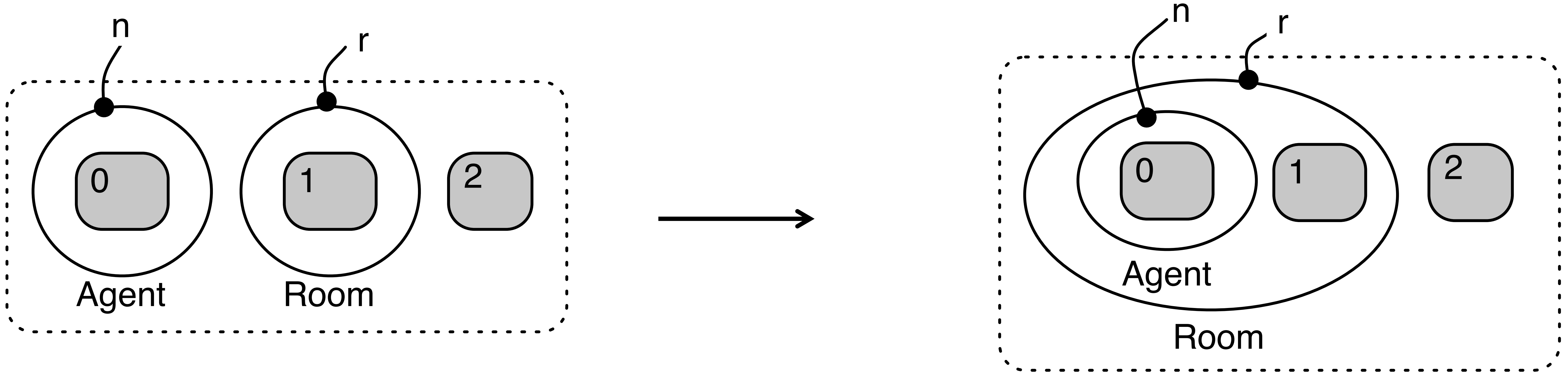
Dynamic Behaviour

Reaction Rules ($R \rightarrow R'$)

A portion of the bigraph matching a redex R is rewritten as the reactum R′.

For Example: Action *enter_room*

Redex (R) Reactum (R′)



$Agent_n.-_0 \mid Room_r.\_{}_1 \mid \_{}_2$ $Room_r.(Agent_n.-_0 \mid \_{}_1) \mid \_{}_2$

Specifying Security Requirements

A topological configuration described by a bigraph *C* satisfies a **property** if the bigraph specifying the property can be matched against *C*.

Example: Violation of the vaccine integrity (SR1)

An RDU transporting the vaccine is in a storage zone that is locked and not cooled

$$Room_x.\left(RDU_y.(Vaccine) \,\middle|\, Locked \right)$$

Security Requirements: Branching Time Temporal Logic (CTL).

$$AG(\neg(SR1))$$

C. Tsigkanos, L. Pasquale, C. Ghezzi and B. Nuseibeh, "On the Interplay Between Cyber and Physical Spaces for Adaptive Security," in *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 3, pp. 466-480, 1 May-June 2018, doi: 10.1109/TDSC.2016.2599880.

Enabling Automated Reasoning

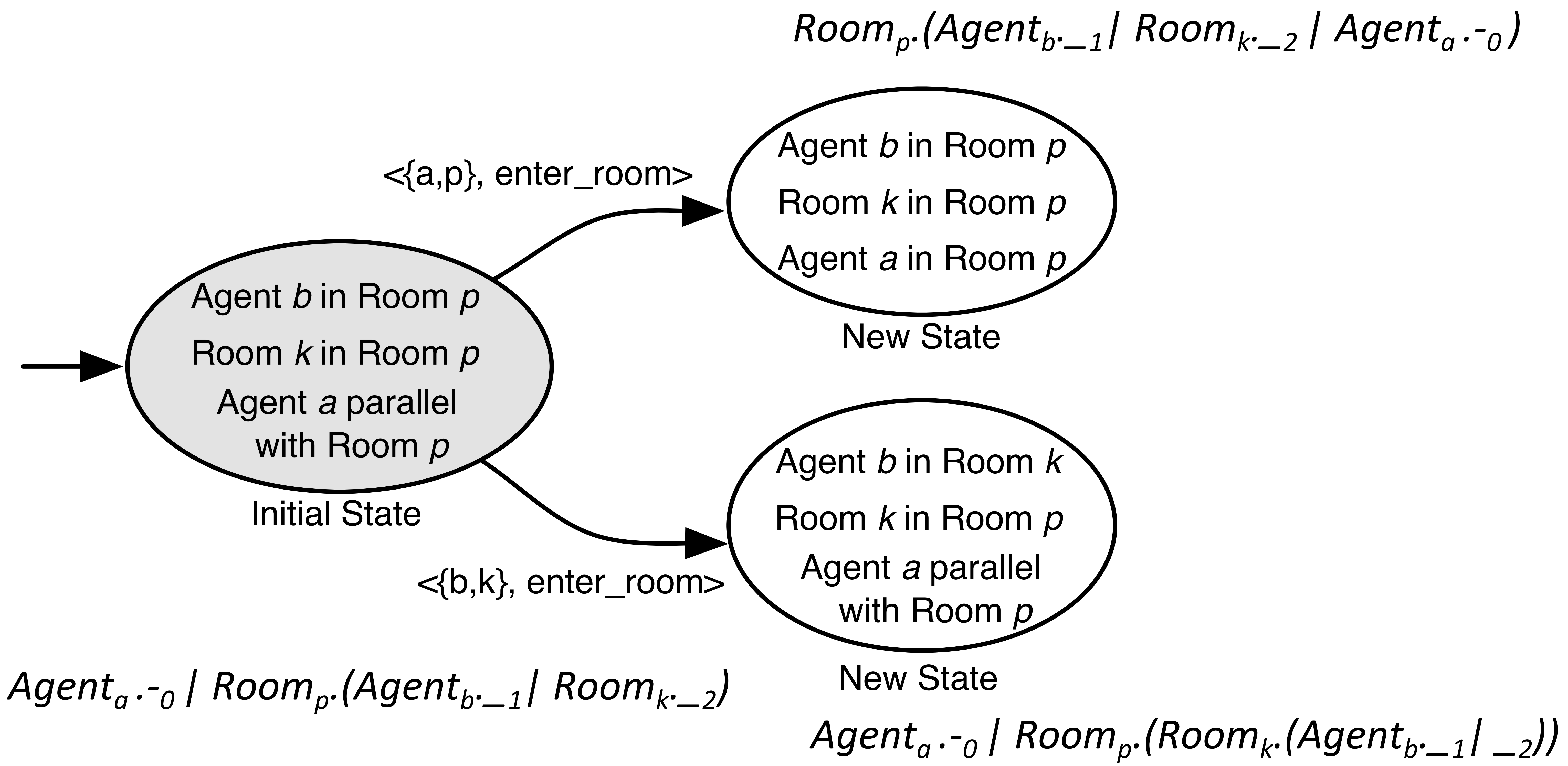
The BRS-based specification is transformed into an equivalent Labelled Transition System (LTS).

Each LTS state represents a different bigraph configuration

Each LTS transition represents a different application of the reaction rules leading to new configurations

C. Tsigkanos, L. Pasquale, C. Ghezzi and B. Nuseibeh, "On the Interplay Between Cyber and Physical Spaces for Adaptive Security," in *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 3, pp. 466-480, 1 May-June 2018, doi: 10.1109/TDSC.2016.2599880.

Enabling Automated Reasoning

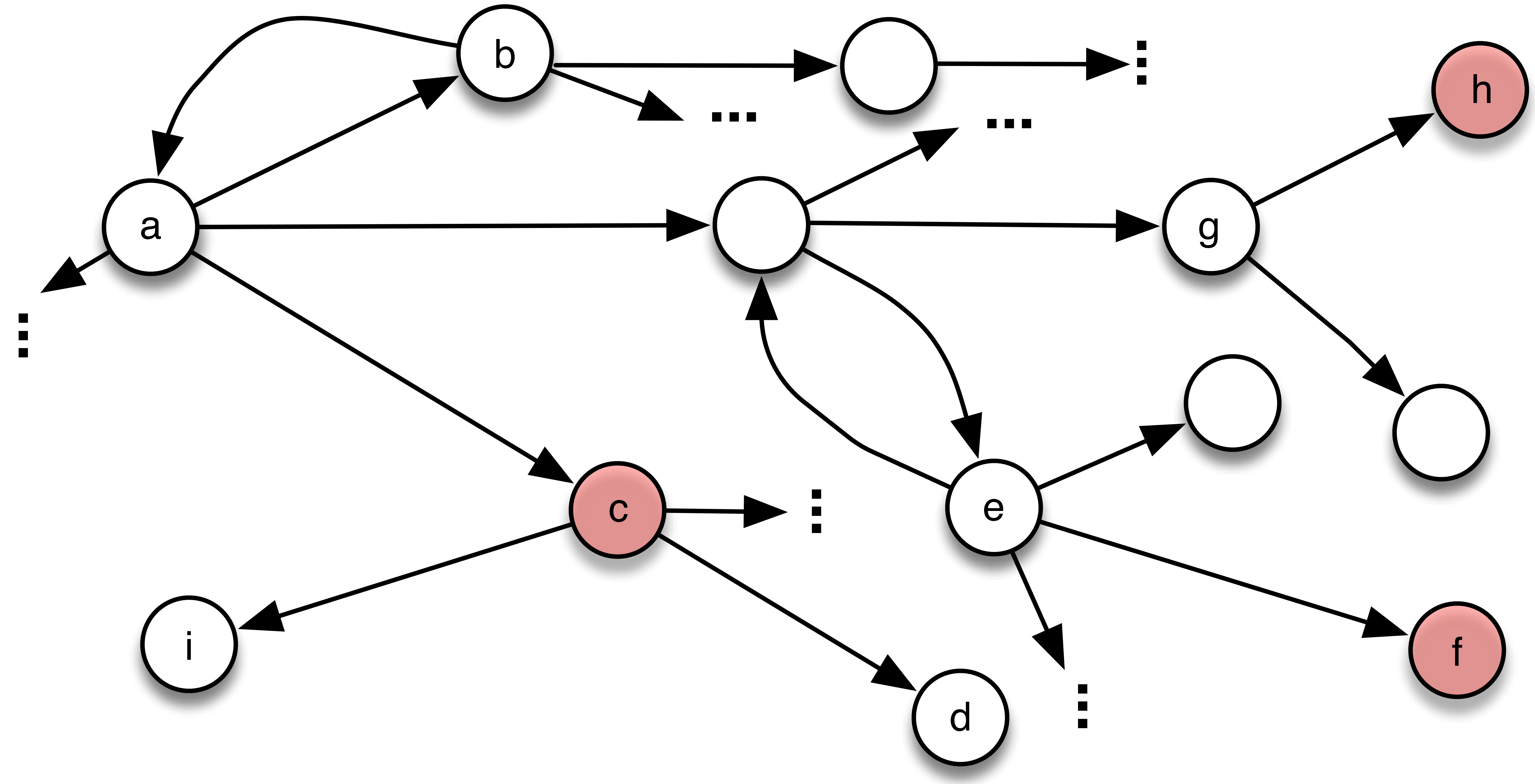


The process is iterated by exploring all configurations and generating new LTS states accordingly.

C. Tsigkanos, L. Pasquale, C. Ghezzi and B. Nuseibeh, "On the Interplay Between Cyber and Physical Spaces for Adaptive Security," in *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 3, pp. 466-480, 1 May-June 2018, doi: 10.1109/TDSC.2016.2599880.

Speculative Threat Analysis

- Identifies potential violations of security requirements that take place in future evolutions of the cyber-physical space.
- Interpret the BRS over an LTS
- Perform explicit state model checking to discover LTS states representing violations

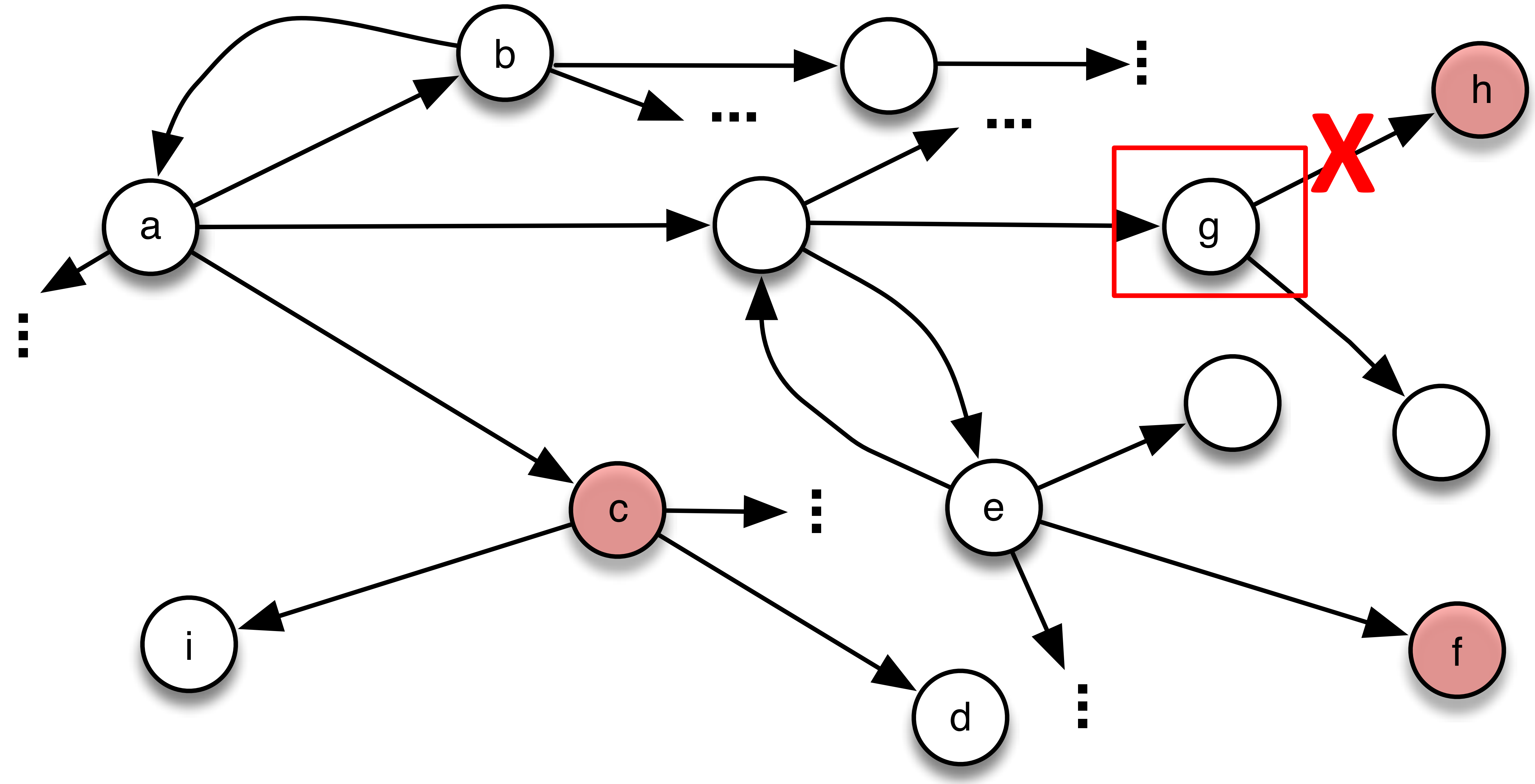


C. Tsigkanos, L. Pasquale, C. Ghezzi and B. Nuseibeh, "On the Interplay Between Cyber and Physical Spaces for Adaptive Security," in *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 3, pp. 466-480, 1 May-June 2018, doi: 10.1109/TDSC.2016.2599880.

Computing Security Controls

When a state (P) immediately before a violating state is entered

- Disable all actions in P leading to a violating states OR

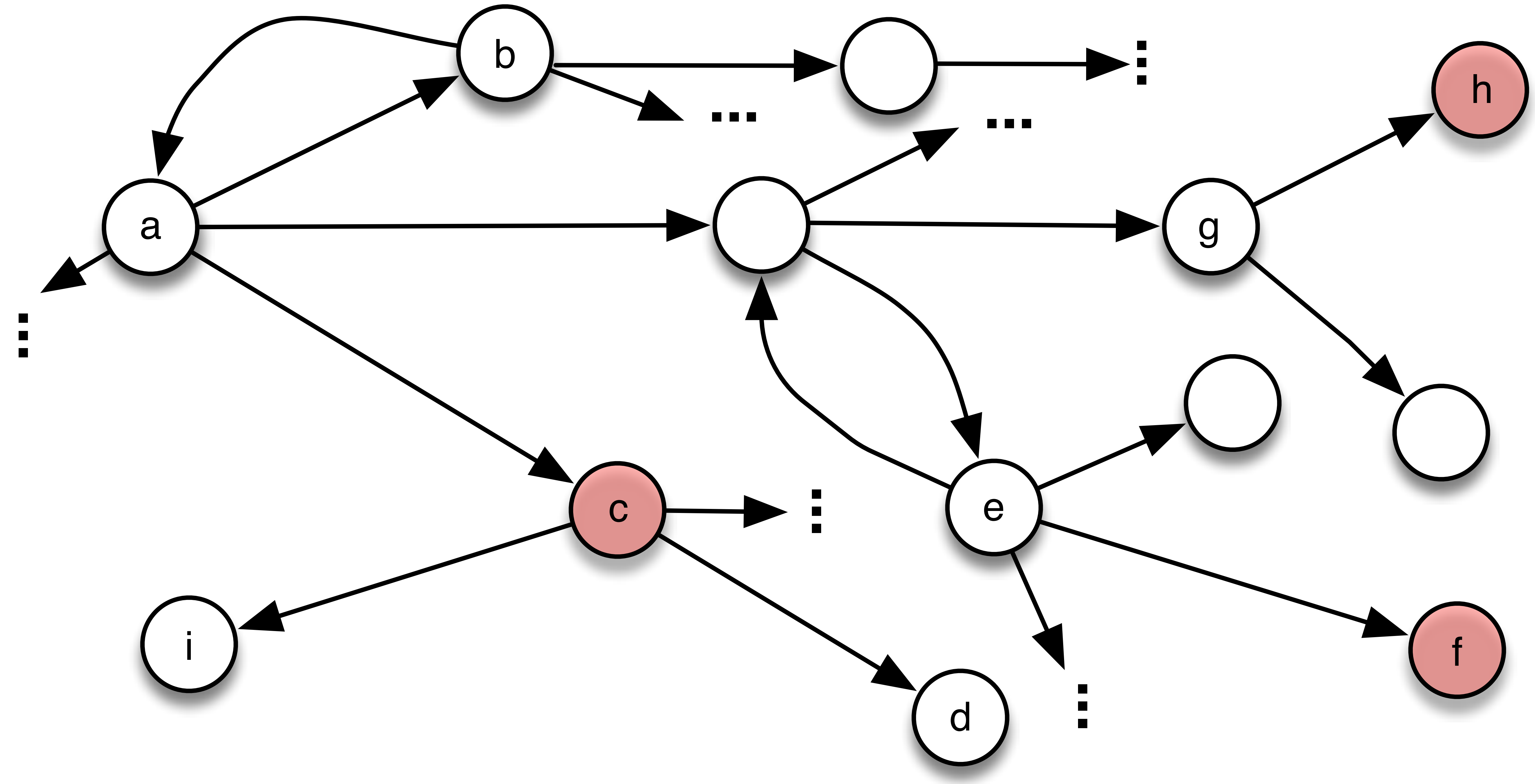


C. Tsigkanos, L. Pasquale, C. Ghezzi and B. Nuseibeh, "On the Interplay Between Cyber and Physical Spaces for Adaptive Security," in *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 3, pp. 466-480, 1 May-June 2018, doi: 10.1109/TDSC.2016.2599880.

Computing Security Controls

When a state (P) immediately before a violating state is entered

- Disable all actions in P leading to a violating states OR
- Enforce the execution of an action that can lead to a safe state

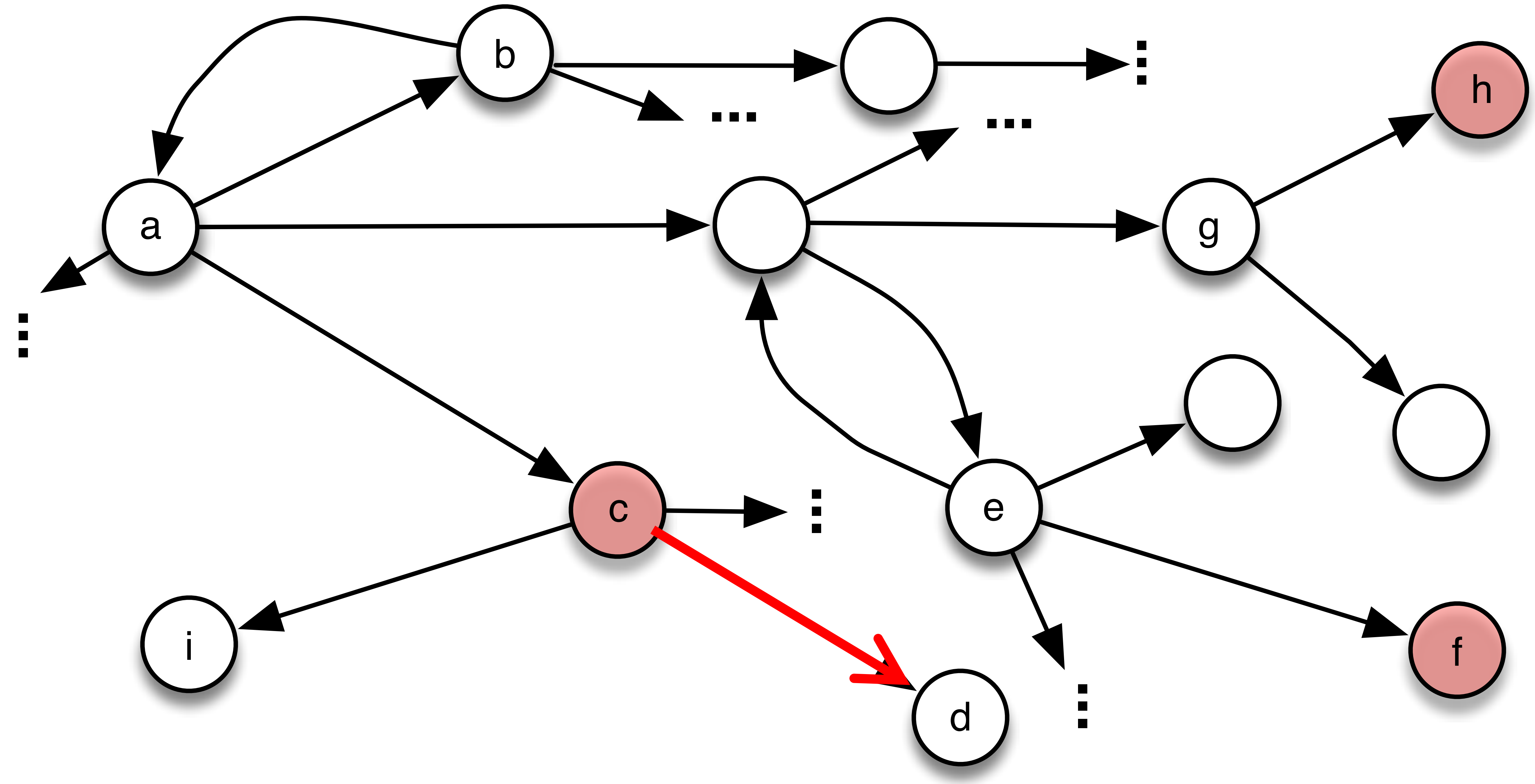


C. Tsigkanos, L. Pasquale, C. Ghezzi and B. Nuseibeh, "On the Interplay Between Cyber and Physical Spaces for Adaptive Security," in *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 3, pp. 466-480, 1 May-June 2018, doi: 10.1109/TDSC.2016.2599880.

Computing Security Controls

When a violating state (V) is reached

- Disable all transitions in V leading to a violating states AND
- Enforce the execution of a transition(s) leading to a safe state

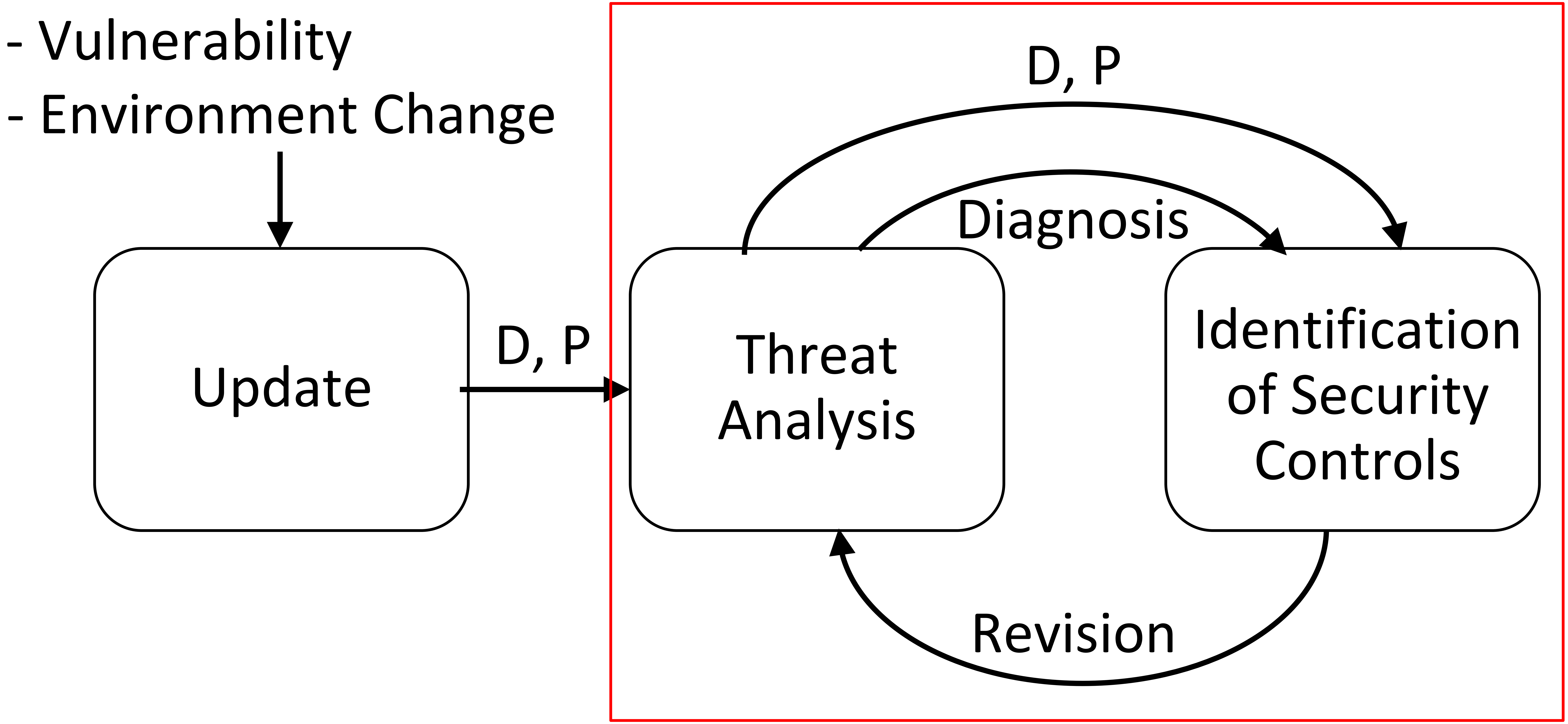


C. Tsigkanos, L. Pasquale, C. Ghezzi and B. Nuseibeh, "On the Interplay Between Cyber and Physical Spaces for Adaptive Security," in *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 3, pp. 466-480, 1 May-June 2018, doi: 10.1109/TDSC.2016.2599880.

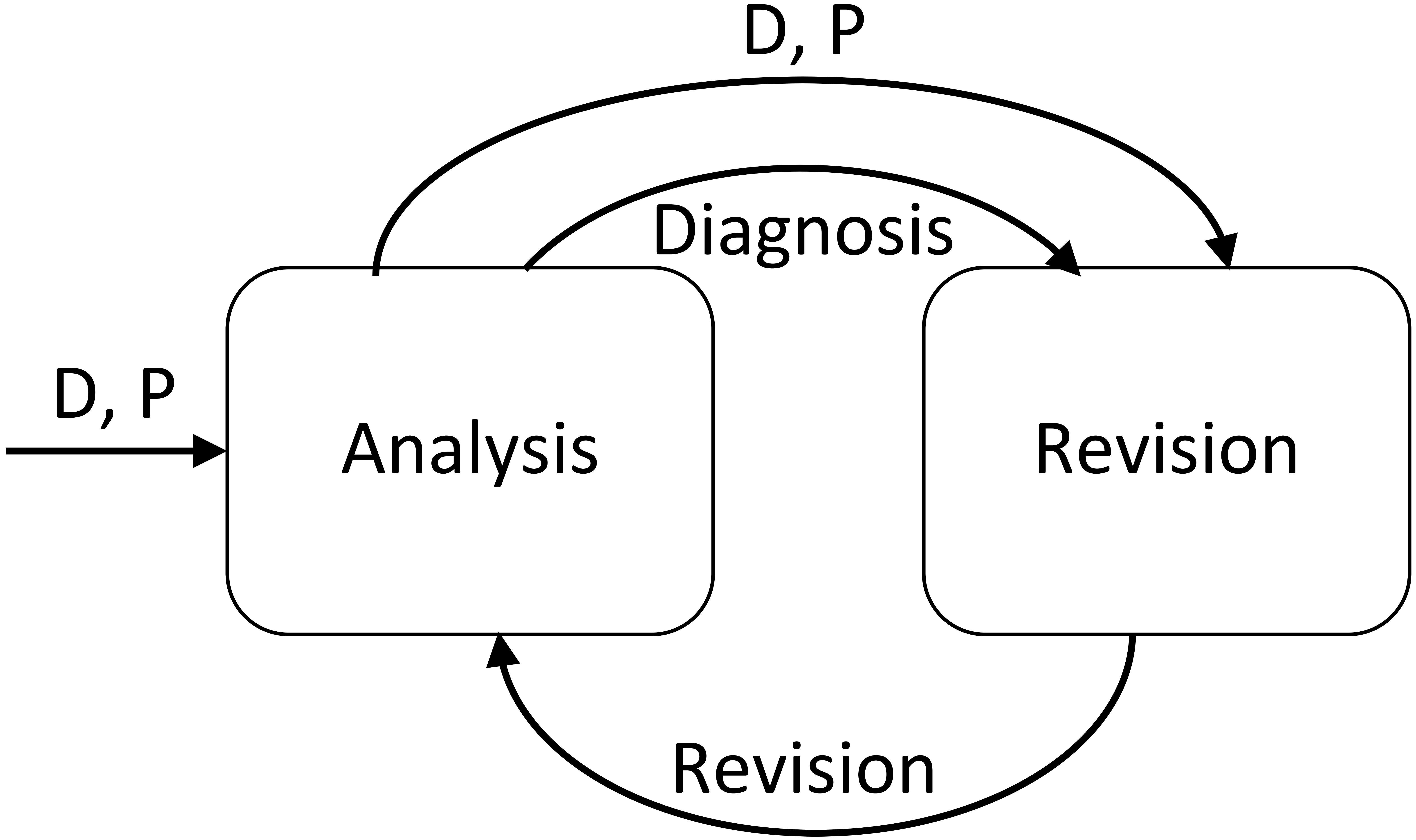
2 3 Key Ideas to Support Sustainable Security

- Formalize and reason about the extended attack surface
- **Discover and counteract new threats**
- Provide explanations to human operators

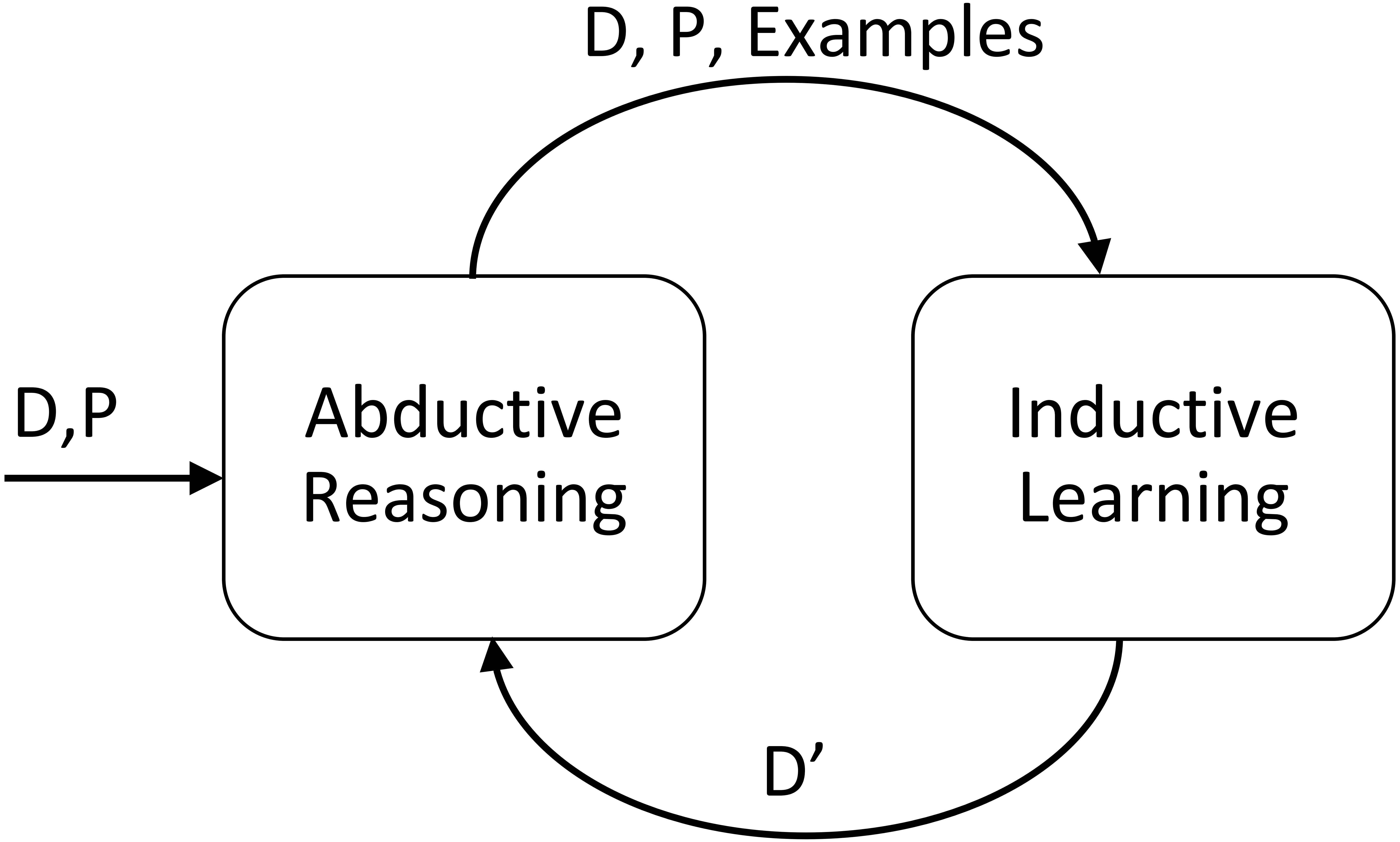
Lifecycle for Handling Unknown Threats



Lifecycle for Handling Unknown Threats

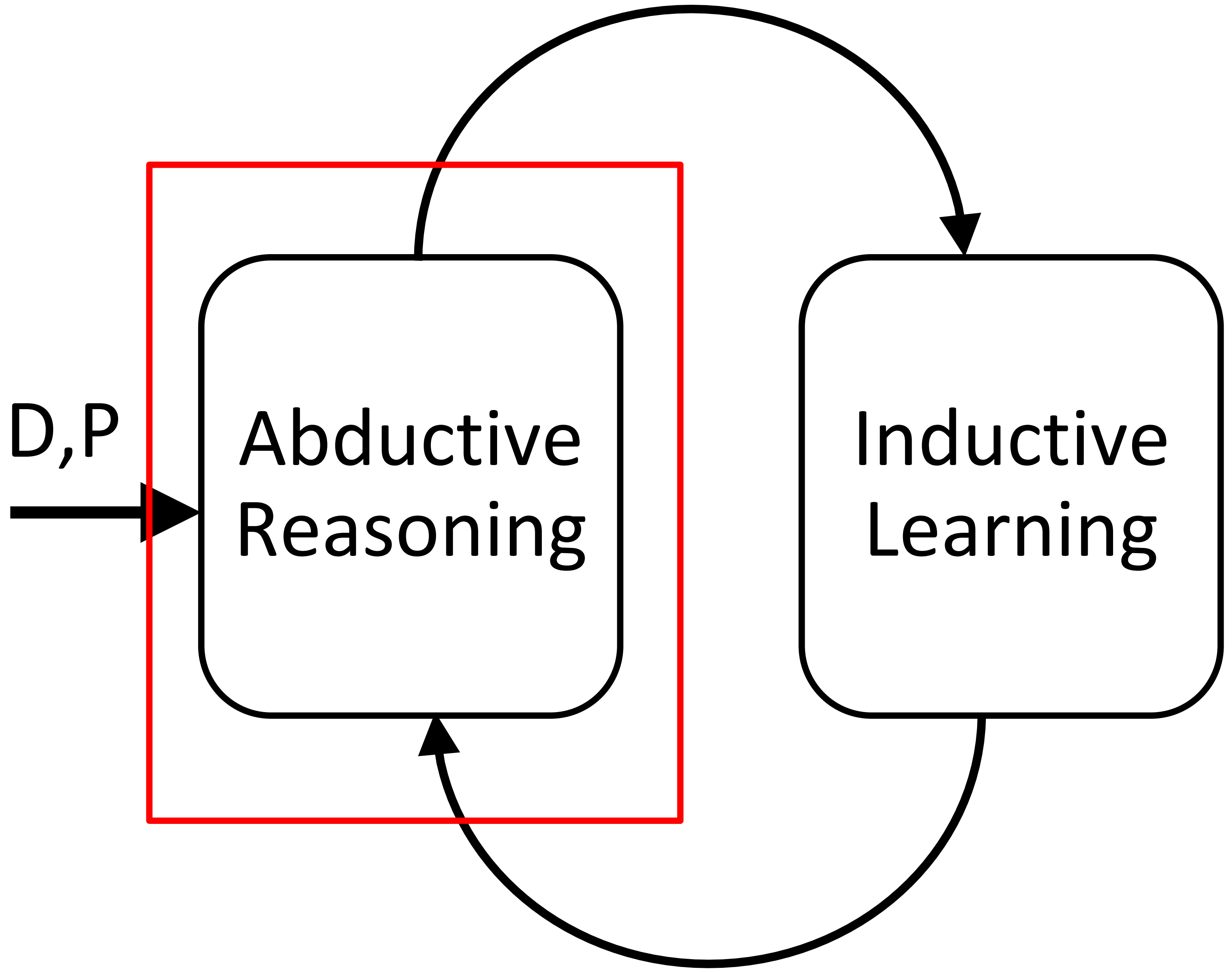


Evolving Requirement Specifications



Garcez, AS d'Avila, Alessandra Russo, Bashar Nuseibeh, and Jeff Kramer. "Combining abductive reasoning and inductive learning to evolve requirements specifications." *IEE Proceedings-Software* 150, no. 1 (2003): 25-38.

Abductive Reasoning

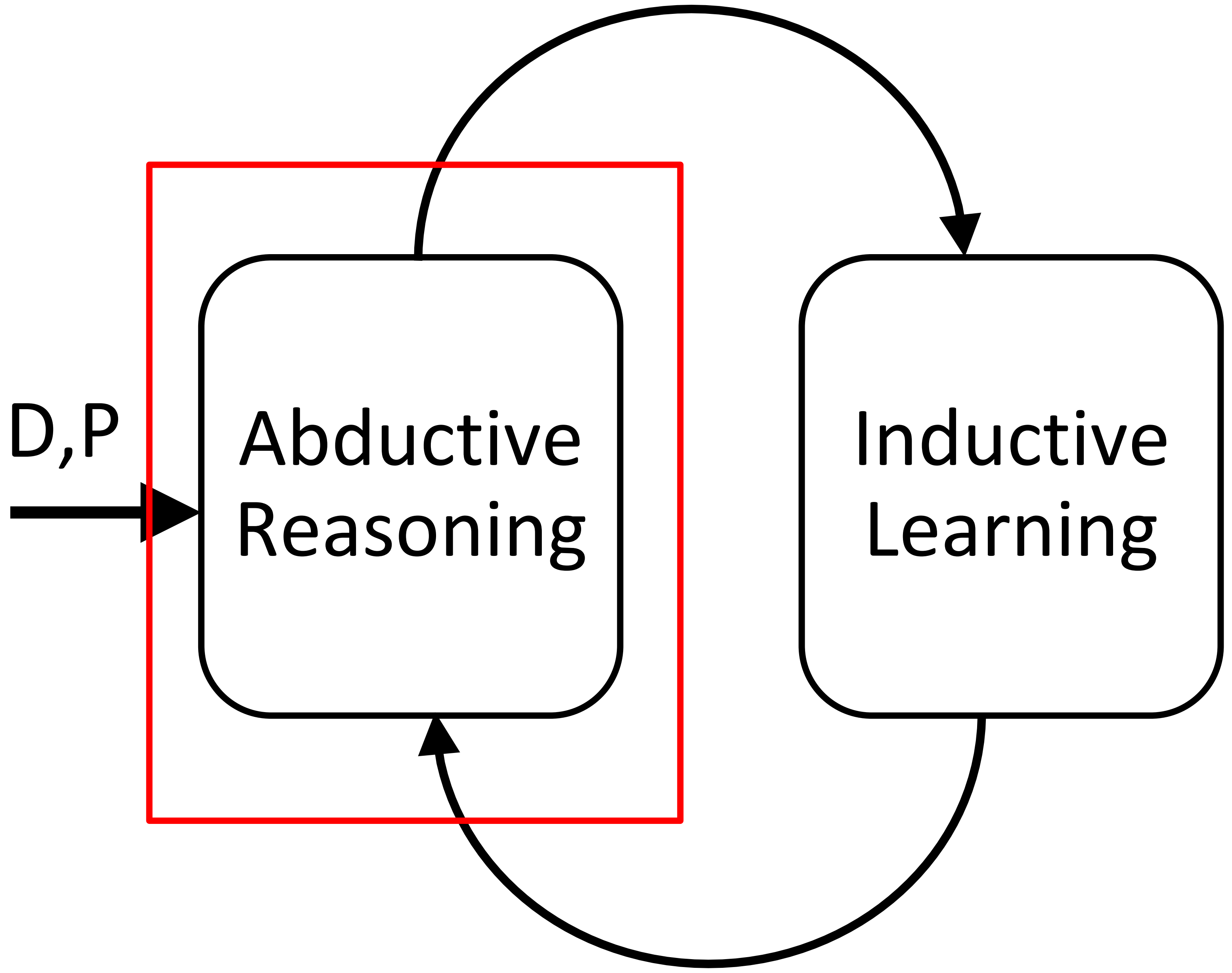


$$D \cup \Delta^{+} \vdash P$$

- In event-driven system descriptions, abduction would be used to identify a trace of events and system transitions (starting from the initial state) that would prove a given requirement.

.

Abductive Learning



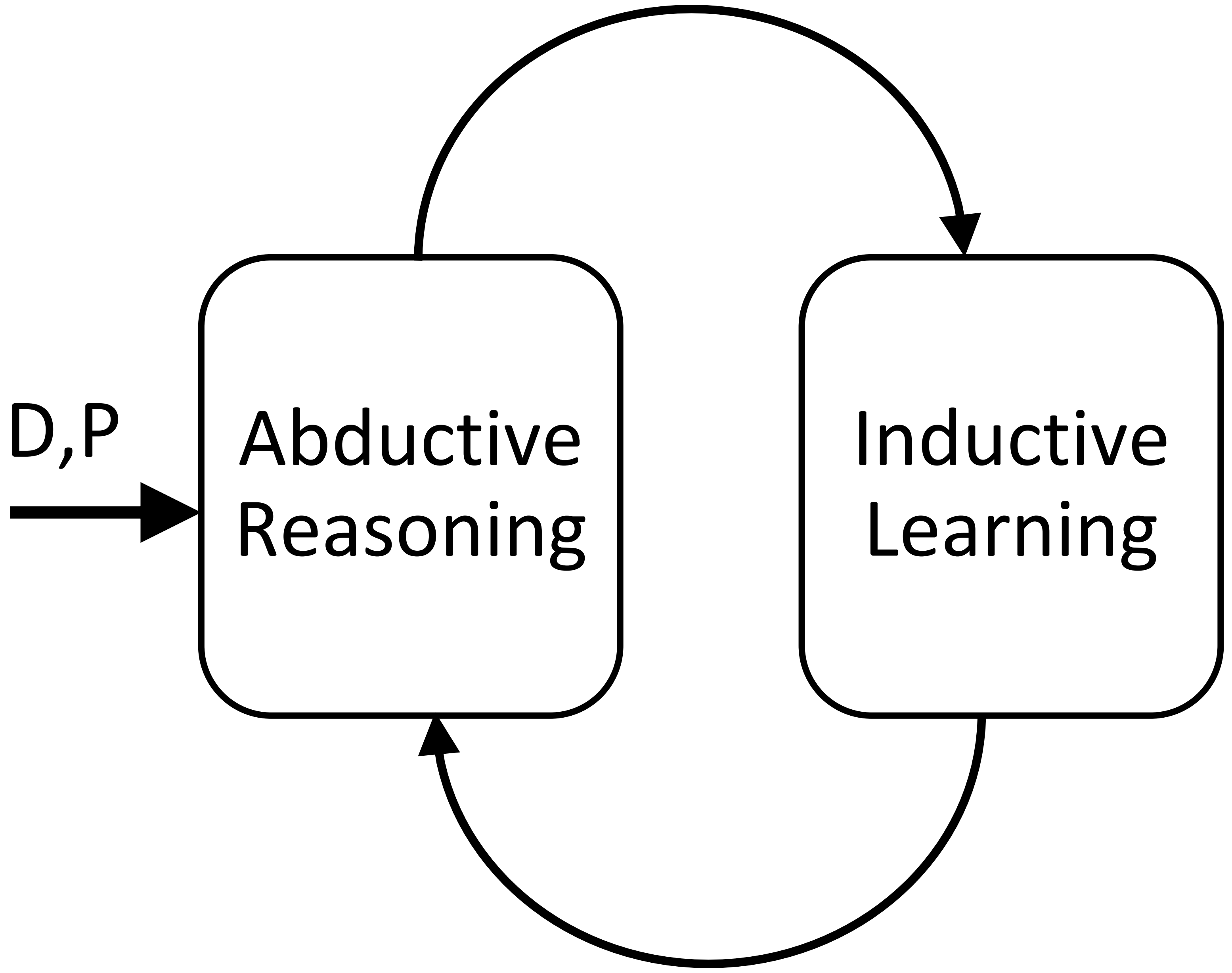
$$D \cup \Delta^{+} \vdash P$$

$$D \cup \Delta^{-} \vdash \neg P$$

- In refutation mode, abduction allows the generation of counter-examples (incorrect system transitions) as diagnostic information of properties violation.

- If the abductive procedure finds such a set (of incorrect state transitions), then acts as a set of counter-examples to the validity of P.

Abductive Learning - Example



$$D \cup \Delta^+ \vdash P$$

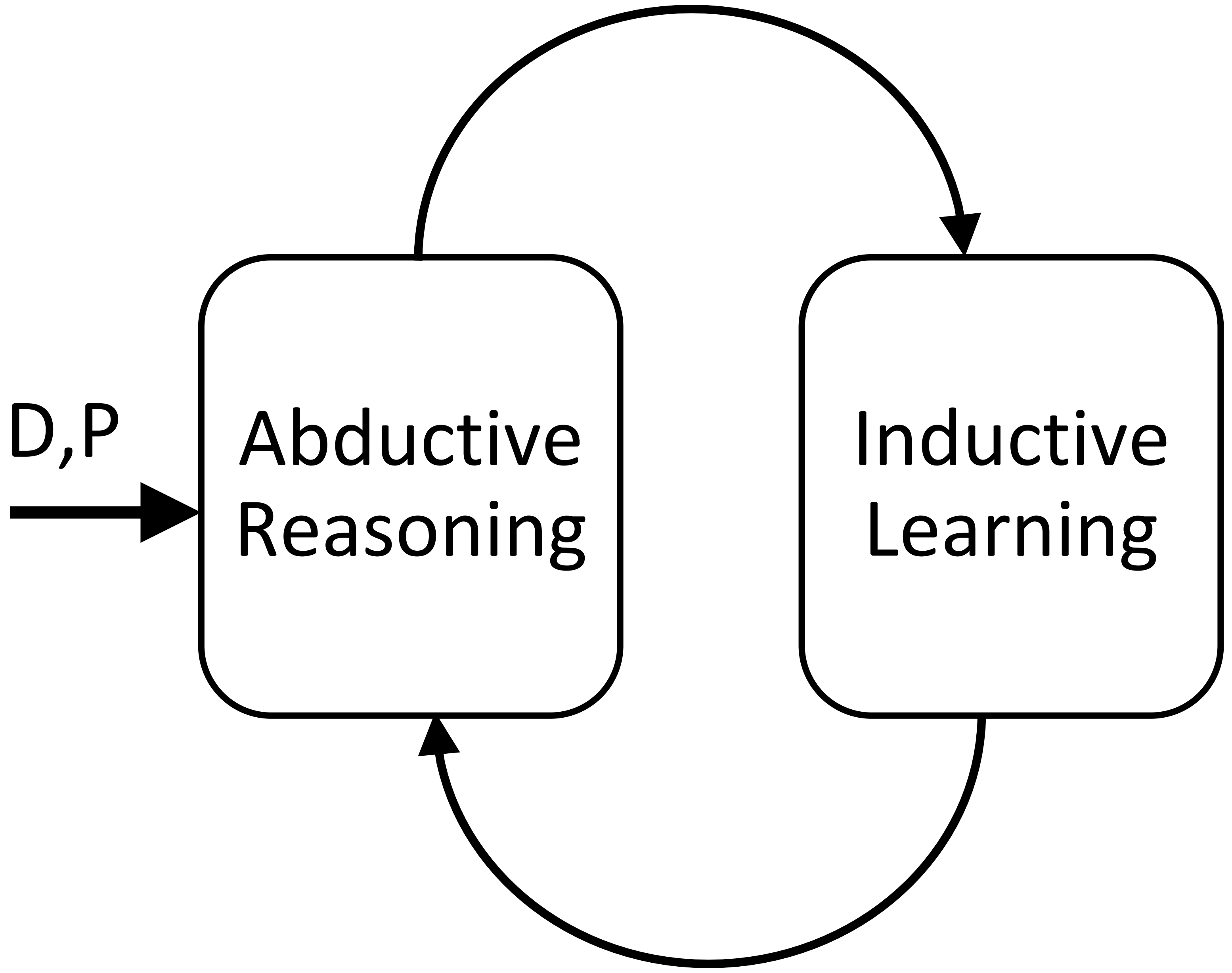
$$D \cup \Delta^- \vdash \neg P$$

Example described above formalized using propositional logic programming:

D: $$In(RDU, Room1) \wedge Connected(Room1, Room2) \wedge Enter(RDU, Room2) \rightarrow In(RDU, Room2)$$

P: $$In(DangMaterial, Room) \rightarrow \neg(RDU, Room)$$

Abductive Learning - Example



$$D \cup \Delta^+ \vdash P$$

$$D \cup \Delta^- \vdash \neg P$$

$$\Delta^+ = \{In(RDU, Room1), Connected(Room1, Room2), Enter(RDU, Room2), In'(RDU, Room2)\}$$

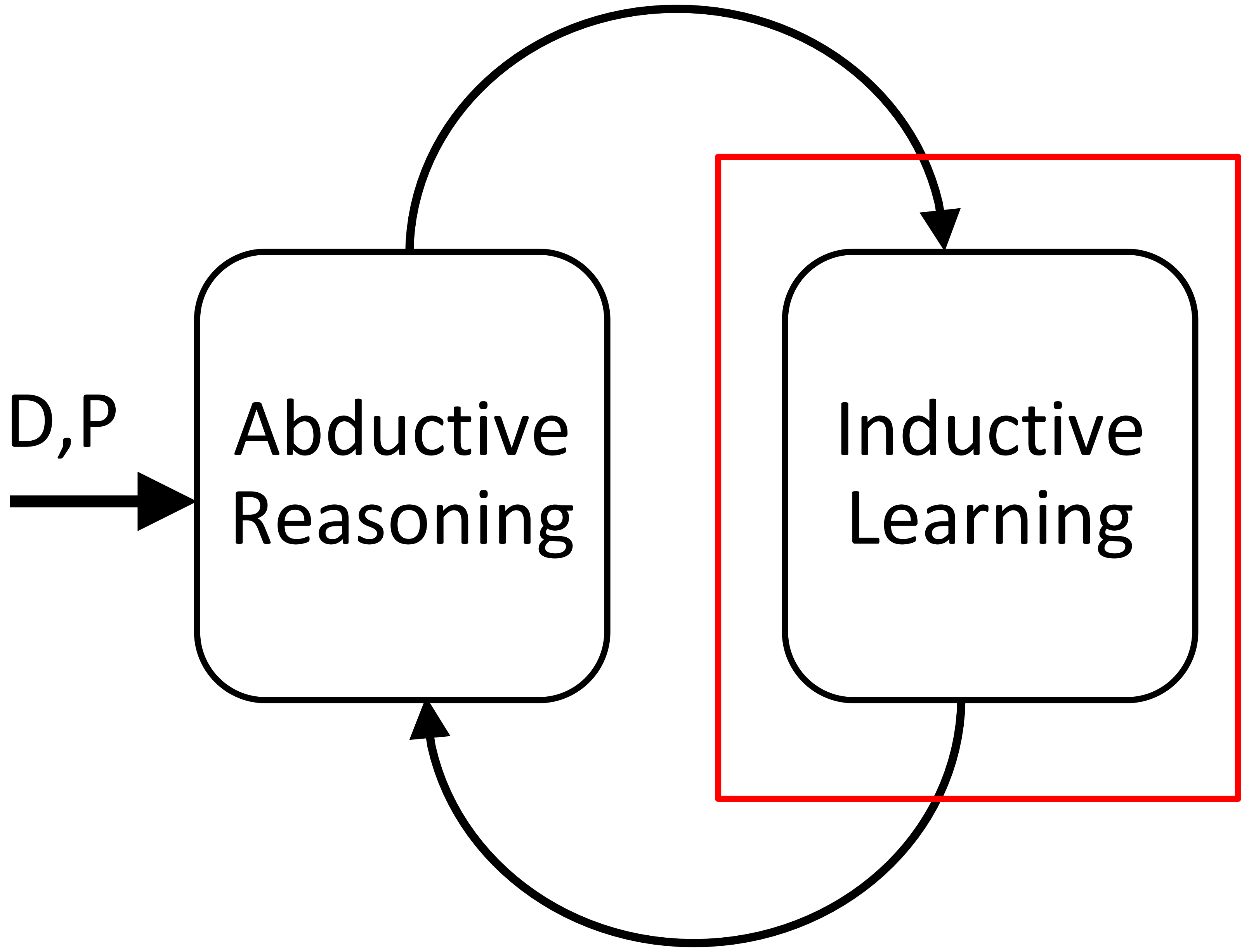
$$\Delta^- = In(RDU, Room1), Connected(Room1, Room2), Enter(RDU, Room2), In(DangMaterial, Room2), In'(RDU, Room2)\}$$

Example described above formalized using propositional logic programming:

D: $In(RDU, Room1) \wedge Connected(Room1, Room2) \wedge Enter(RDU, Room2) \rightarrow In(RDU, Room2)$

P: $In(DangMaterial, Room) \rightarrow \neg(RDU, Room)$

Inductive Learning

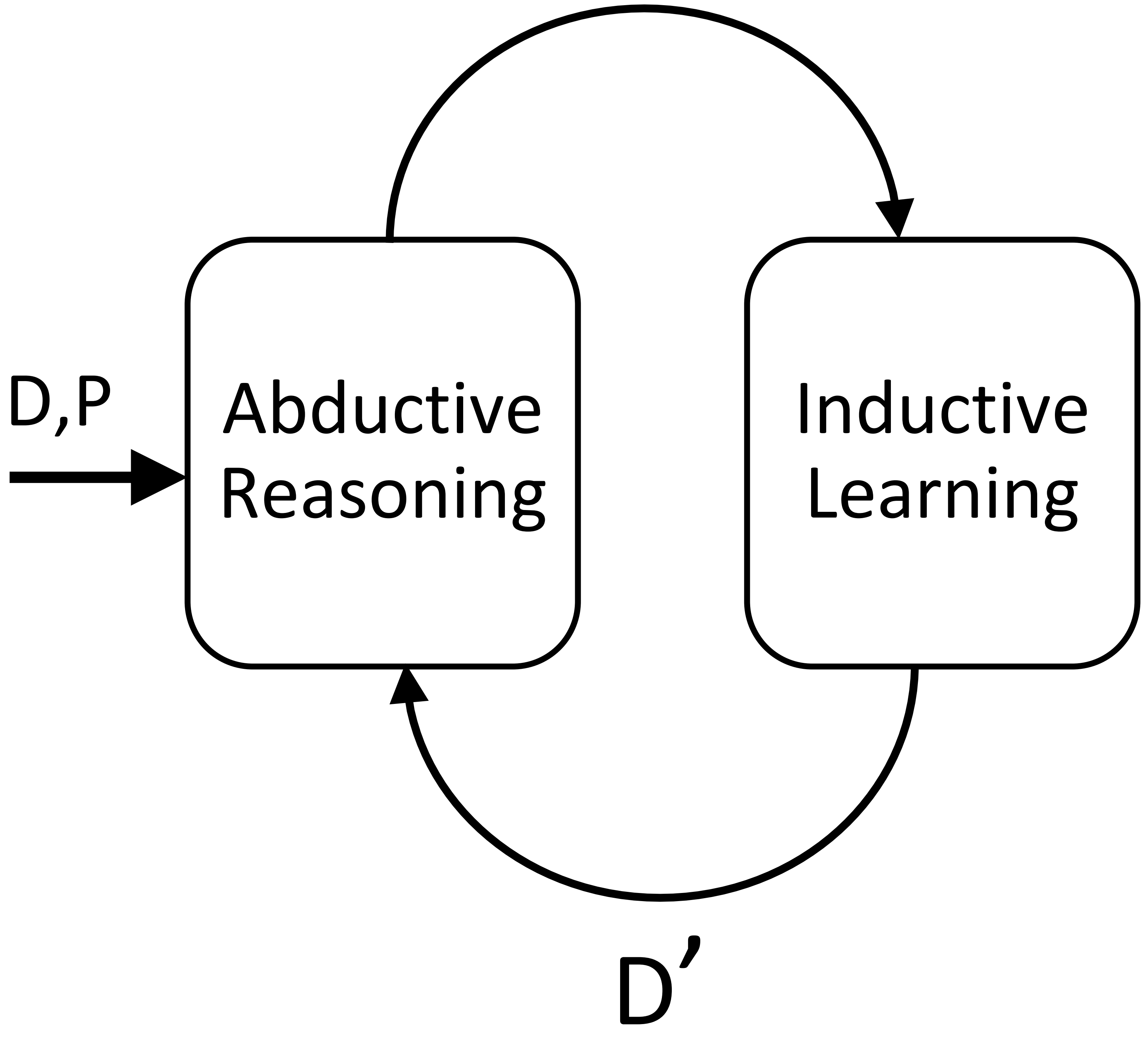


$$D \cup h \vdash \Delta^{+}$$

$$D \cup h \nvdash \Delta^{-}$$

Aims to find hypotheses, in the form of rules, that are consistent with the description of the system (background knowledge) to explain a given set of examples.

Inductive Learning - Example



$$D \cup h \vdash \Delta^+$$

$$D \cup h \nvdash \Delta^-$$

$$\Delta^+ = \{In(RDU, Room1), Connected(Room1, Room2), Enter(RDU, Room2), In'(RDU, Room2)\}$$

$$\Delta^- = In(RDU, Room1), Connected(Room1, Room2), Enter(RDU, Room2), In(DangMaterial, Room2), In'(RDU, Room2)\}$$

D': $In(RDU, Room1) \wedge Connected(Room1, Room2) \wedge Enter(RDU, Room2) \wedge \neg In(DangMaterial, Room2) \rightarrow In(RDU, Room2)$

2 3 Key Ideas to Support Sustainable Security

- Formalize and reason about the extended attack surface
- Discover and counteract new threats
- **Provide explanations to human operators**

Objective

Providing explanations about why a system produces a certain behaviour

- E.g., justifying why a certain security requirement is violated

Providing a human operator with a full model of the system, the operating environment and their current state is infeasible due to information overload.

Abstraction is Key

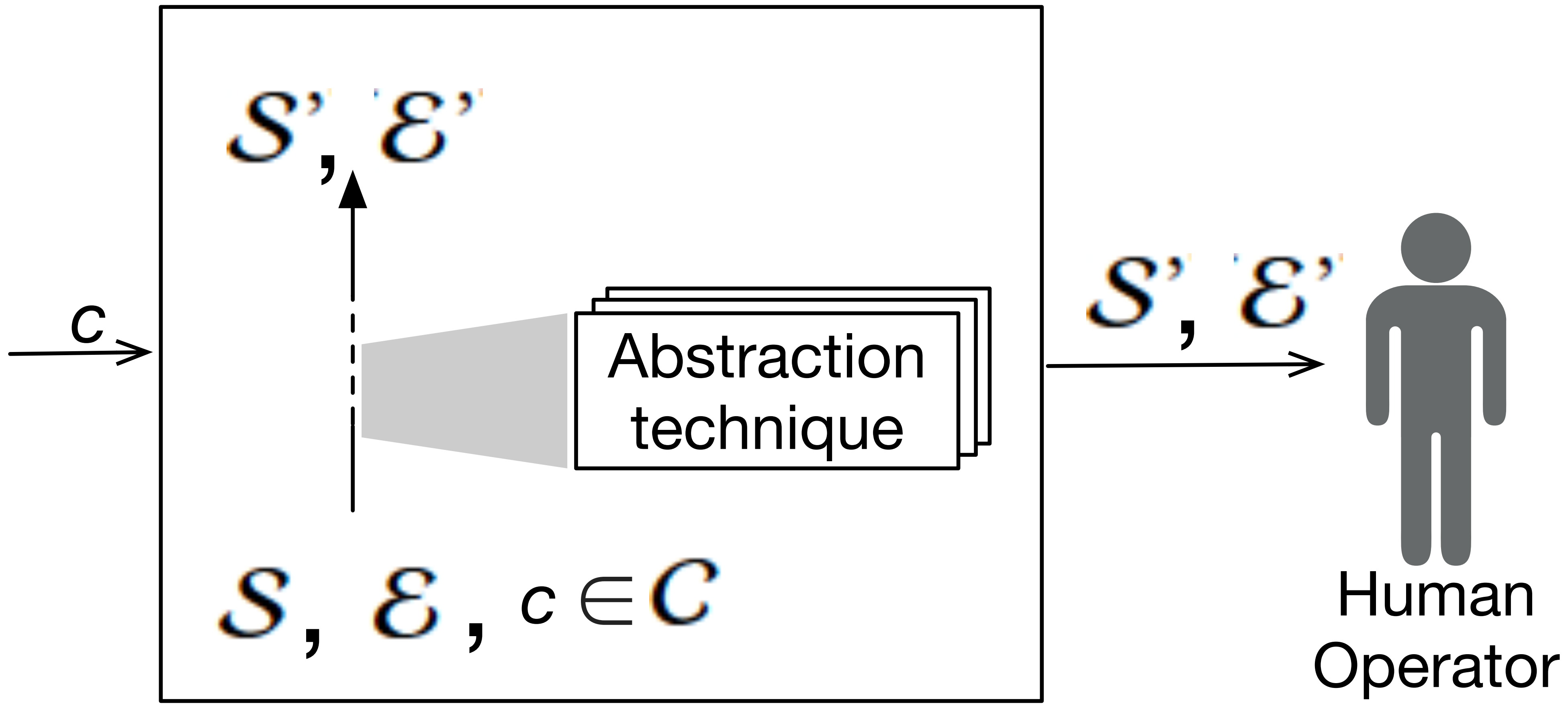
Abstraction can be used to reduce the complexity of the model

- Provide a high level of detail for the aspects of the system that affect satisfaction of some requirements of interest, while leaving irrelevant aspects under-specified.
- Aligned with the principles of Situation Awareness Oriented Design
- Can facilitate decision-making.
- Abstraction has been used in previous work to provide explanations about machine interfaces [Combefis et al. 2011]

L. Pasquale, V. Gervasi, "On the Use of Abstractions to Provide Explanations," submitted at the ESEC/FSE 2021 IVR Track.

Abstraction is Key

Abstraction can be used to reduce the complexity of the model



- Provide a high level of detail for the aspects of the system that affect satisfaction of some requirements of interest, while leaving irrelevant aspects under-specified.
- Abstraction has been used in previous work to provide explanations about machine interfaces [Combefis et al. 2011]

L. Pasquale, V. Gervasi, "On the Use of Abstractions to Provide Explanations," submitted at the ESEC/FSE 2021 IVR Track.

Example – Abstract State Machines (ASM)

- **ASM**: A set of rules that are conditioned by, and may generate updates for, states.
- **State =>** defined as a set of locations
- **Location =>** identified by a function symbol and a list of parameters associated with values

Example – Abstract State Machines (ASM)

- **ASM**: A set of rules that are conditioned by, and may generate updates for, states.
- **State =>** defined as a set of locations
- **Location =>** identified by a function symbol and a list of parameters associated with values

System Model

Concrete System Model

forall r **in** *Rooms*
if $curtemp(r) > destemp(r) + h$ **then** $aircond(r) :=$ on
if $curtemp(r) < destemp(r) - h$ **then** $aircond(r) :=$ off
if $curtemp(r) > destemp(r) + k$ **then** $heating(r) :=$ off
if $curtemp(r) < destemp(r) - k$ **then** $heating(r) :=$ on
if $presence(r)$ **then** $lights(r) :=$ on

Example – Abstract State Machines (ASM)

- **ASM**: A set of rules that are conditioned by, and may generate updates for, states.
- **State =>** defined as a set of locations
- **Location =>** identified by a function symbol and a list of parameters associated with values

System Model

Concrete System Model

forall r **in** *Rooms*
if $curtemp(r) > destemp(r) + h$ **then** $aircond(r) := \text{on}$
if $curtemp(r) < destemp(r) - h$ **then** $aircond(r) := \text{off}$
if $curtemp(r) > destemp(r) + k$ **then** $heating(r) := \text{off}$
if $curtemp(r) < destemp(r) - k$ **then** $heating(r) := \text{on}$
if $presence(r)$ **then** $lights(r) := \text{on}$

Constraints

$c_1)$ $\forall r \in Rooms, \neg(aircond(r) = \text{on} \wedge heating(r) = \text{on})$

$c_2)$ $\forall r \in Rooms, window(r) = \text{open} \Rightarrow (aircond(r) = \text{off} \wedge heating(r) = \text{off})$

$c_3)$ $\forall r \in Rooms \setminus Halls, \neg presence(r) \Rightarrow lights(r) = \text{off}$

Focus on Interesting Variables

c_1) $\forall r \in Rooms, \neg(aircond(r) = \mathsf{on} \wedge heating(r) = \mathsf{on})$

Concrete System Model

forall r **in** $Rooms$
 if $curtemp(r) > destemp(r) + h$ **then** $aircond(r) := \mathsf{on}$
 if $curtemp(r) < destemp(r) - h$ **then** $aircond(r) := \mathsf{off}$
 if $curtemp(r) > destemp(r) + k$ **then** $heating(r) := \mathsf{off}$
 if $curtemp(r) < destemp(r) - k$ **then** $heating(r) := \mathsf{on}$
 if $presence(r)$ **then** $lights(r) := \mathsf{on}$

⬇

Step 1

forall r **in** $\{q\}$
 if $curtemp(r) > destemp(r) + h \ \ldots$

Focus on Interesting Locations

c_1) $\forall r \in Rooms, \neg(aircond(r) = \mathsf{on} \wedge heating(r) = \mathsf{on})$

The only locations affecting the violated constraint are aircond(q) and heating(q)

Step 1

forall r **in** $\{q\}$
if $curtemp(r) > destemp(r) + h \ \ldots$

⬇

Step 2

if $curtemp(q) > destemp(q) + h$ **then** $aircond(q) := \mathsf{on}$
if $curtemp(q) < destemp(q) - h$ **then** $aircond(q) := \mathsf{off}$
if $curtemp(q) > destemp(q) + k$ **then** $heating(q) := \mathsf{off}$
if $curtemp(q) < destemp(q) - k$ **then** $heating(q) := \mathsf{on}$

Focus on Interesting Value

$c_1)\ \forall r \in Rooms, \neg(aircond(r) = \mathsf{on} \land heating(r) = \mathsf{on})$

The values of aircond(r) and heating(r) are both on

We can splice the model considering 3 sets of rules:

- **R1**: those that set one of the interesting locations to the value observed in the state at time of violation
- **R2**: those that set one of the interesting locations to a value different from what has been observed in the state at the time of violation
- **R3:** those that do not update the interesting locations

Step 2

if $curtemp(q) > destemp(q) + h$ **then** $aircond(q) := \mathsf{on}$
if $curtemp(q) < destemp(q) - h$ **then** $aircond(q) := \mathsf{off}$
if $curtemp(q) > destemp(q) + k$ **then** $heating(q) := \mathsf{off}$
if $curtemp(q) < destemp(q) - k$ **then** $heating(q) := \mathsf{on}$

Focus on Interesting Value

$c_1)\ \forall r \in Rooms, \neg(aircond(r) = \text{on} \wedge heating(r) = \text{on})$

The values of aircond(r) and heating(r) are both on

We can splice the model considering 3 sets of rules:

- **R1**: those that set one of the interesting locations to the value observed in the state at time of violation
- **R2**: those that set one of the interesting locations to a value different from what has been observed in the state at the time of violation
- **R3:** those that do not update the interesting locations

Step 2

$$
\begin{aligned}
&\textbf{if } curtemp(q) > destemp(q) + h \textbf{ then } aircond(q) := \text{on}\\
&\textbf{if } curtemp(q) < destemp(q) - h \textbf{ then } aircond(q) := \text{off}\\
&\textbf{if } curtemp(q) > destemp(q) + k \textbf{ then } heating(q) := \text{off}\\
&\textbf{if } curtemp(q) < destemp(q) - k \textbf{ then } heating(q) := \text{on}
\end{aligned}
$$

Focus on Interesting Value

$c_1)\ \forall r \in Rooms, \neg(aircond(r) = \mathsf{on} \wedge heating(r) = \mathsf{on})$

The values of aircond(r) and heating(r) are both on

We can splice the model considering 3 sets of rules:

- **R1**: those that set one of the interesting locations to the value observed in the state at time of violation
- **R2**: those that set one of the interesting locations to a value different from what has been observed in the state at the time of violation
- ~~**R3:** those that do not update the interesting locations~~

Step 2

$$\begin{array}{l}
\textbf{if } curtemp(q) > destemp(q) + h \textbf{ then } aircond(q) := \mathsf{on} \\
\textbf{if } curtemp(q) < destemp(q) - h \textbf{ then } aircond(q) := \mathsf{off} \\
\textbf{if } curtemp(q) > destemp(q) + k \textbf{ then } heating(q) := \mathsf{off} \\
\textbf{if } curtemp(q) < destemp(q) - k \textbf{ then } heating(q) := \mathsf{on}
\end{array}$$

Focus on Interesting Value

$c_1)\ \forall r \in Rooms, \neg(aircond(r) = \mathrm{on} \wedge heating(r) = \mathrm{on})$

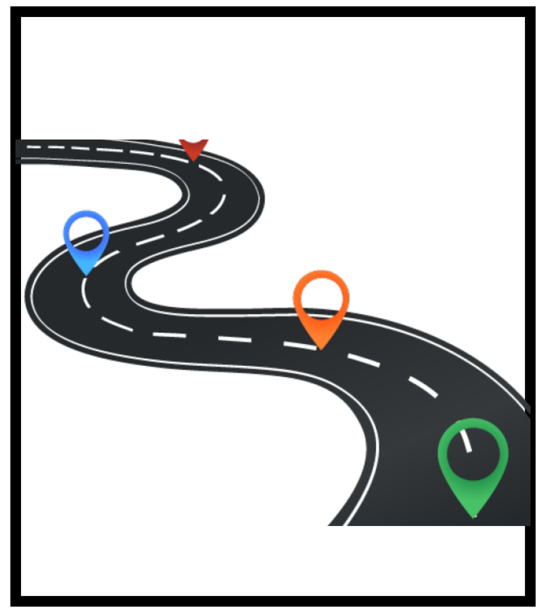
The values of aircond(r) and heating(r) are both on

We can splice the model considering 3 sets of rules:

- **R1**: those that set one of the interesting locations to the value observed in the state at time of violation
- **R2**: those that set one of the interesting locations to a value different from what has been observed in the state at the time of violation
- ~~**R3:** those that do not update the interesting locations~~

Step 3

$$curtemp(q) > destemp(q) + h$$
$$curtemp(q) < destemp(q) - k$$
$$curtemp(q) \geq destemp(q) - h$$
$$curtemp(q) \leq destemp(q) + k$$



3

A Sustainable Security Roadmap

Formalize and reason about the extended attack surface

- Exploit possible compositionality of security properties

- Explore model-based diagnosis techniques (e.g., hierarchical diagnosis [Mozetič, 1991][Siddiqi, 2007]), widely explored in control theory to reduce the computational complexity of the diagnosis.

Discover and Counteract New Threats

Continuous Threat Analysis

Provide Explanations to Human Operators

- Formalize and create a collection of abstraction strategy that can be systematically selected to provide explanations

- Select a suitable level of abstraction depending on:
    - the cognitive abilities of the human operator
    - the time available to a human operator to make a decision

References

1. Venters, Colin C., Norbert Seyff, Christoph Becker, Stefanie Betz, Ruzanna Chitchyan, Leticia Duboc, Dan McIntyre, and Birgit Penzenstadler. "Characterising sustainability requirements: A new species red herring or just an odd fish?." In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS)*, pp. 3-12. IEEE, 2017.

2. Becker, Christoph, Ruzanna Chitchyan, Leticia Duboc, Steve Easterbrook, Birgit Penzenstadler, Norbert Seyff, and Colin C. Venters. "Sustainability design and software: The karlskrona manifesto." In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 2, pp. 467-476. IEEE, 2015.

3. L. Cardelli and A. D. Gordon, "Mobile Ambients," in *Proc. of the 1st Int. Conf. on Foundations of Software Science and Computation Structure*, 1998, pp. 140–155.

4. R. Milner, *"The Space and Motion of Communicating Agents,"* Cambridge University Press, 2009.

5. Tsigkanos, Christos, Liliana Pasquale, Claudio Menghi, Carlo Ghezzi, and Bashar Nuseibeh. "Engineering topology aware adaptive security: Preventing requirements violations at runtime." In *Proc of the 22nd InternationalRequirements Engineering Conference*, pp. 203-212, 2014.

6. Combéfis, Sébastien, Dimitra Giannakopoulou, Charles Pecheur, and Michael Feary. "Learning system abstractions for human operators." In *Proceedings of the international workshop on machine learning technologies in software engineering*, pp. 3-10. 2011.

7. Igor Mozetič. 1991. Hierarchical Model-based Diagnosis. Int. Journal of Man- Machine Studies 35, 3 (1991), 329–362

8. Sajjad Ahmed Siddiqi, Jinbo Huang, et al. 2007. Hierarchical Diagnosis of Multiple Faults. In IJCAI, Vol. 7. 581–586.

THANK YOU!