# SECURITY REQUIREMENTS IN STSs

Mattia Salnitri

mattia.salnitri@unitn.it

# OUTLINE

- What are STSs
- Linking security requirements in STSs
- Checking alignment of security requirements
- Conclusion

# SOCIO-TECHNICAL SYSTEMS (STSs)

- STS are **information systems** where **human and technical components interact** to each others so to achieve common objectives.
- Examples:
  - Health Care System (HCS)
  - Smart cities
  - Smart houses
  - Air traffic management
  - E-government system

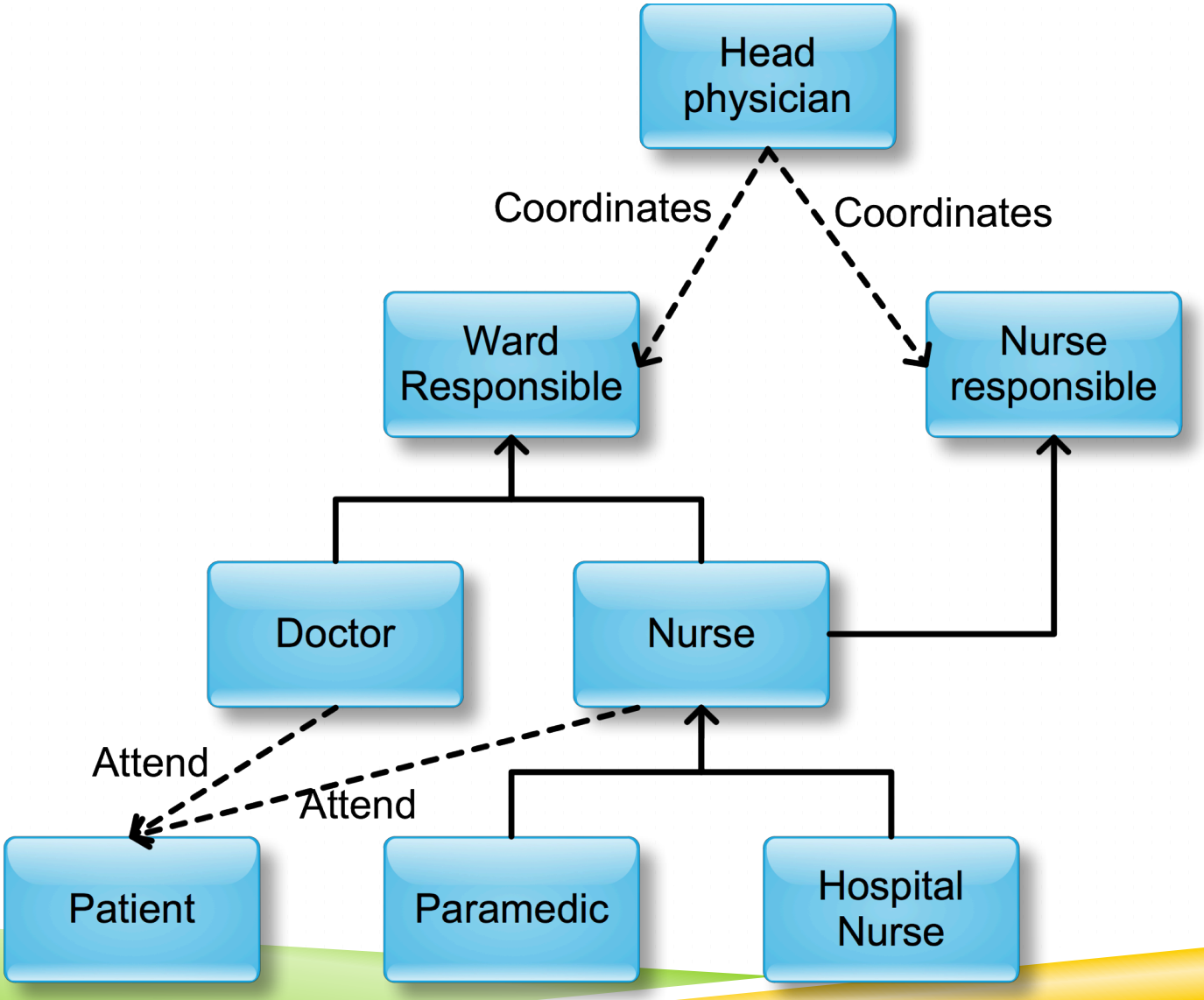
# LINKING SECURITY REQUIREMENTS IN STSs

# STS REQUIREMENTS

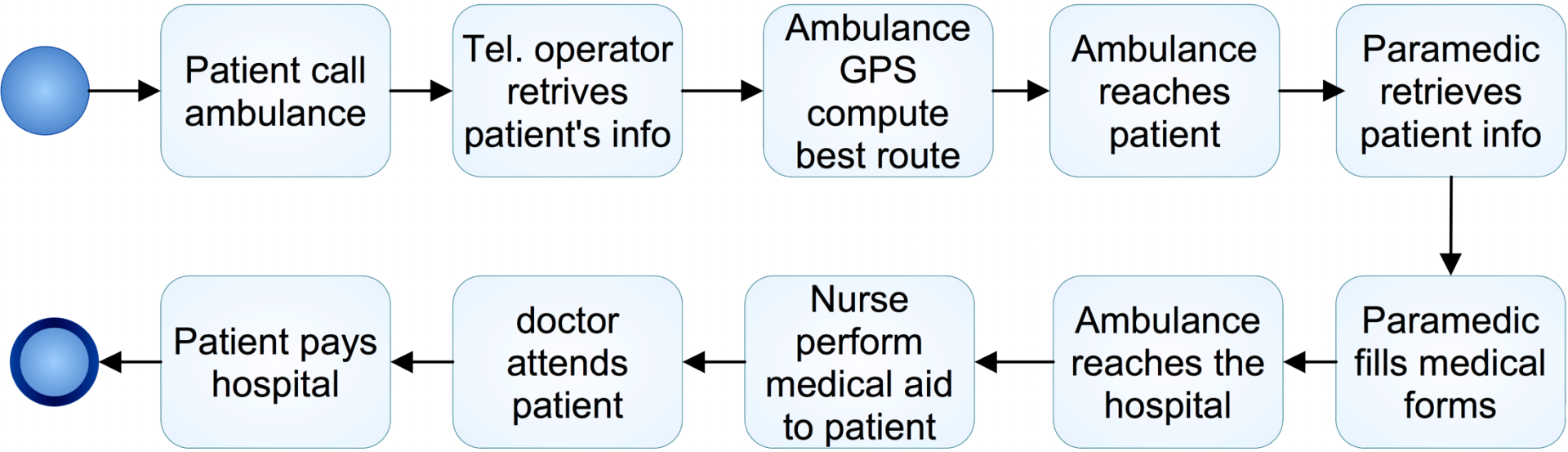
- Organization structure
  - The typically hierarchical arrangement of lines of authority, communications, rights and duties of an organization[1].
- Processes
  - A collection of activities that takes one or more kinds of input and creates an output[6].
- Technologies
  - Hardware and software functional to the execution of activities in processes

# HCS EXAMPLE - SANTA CHIARA HOSPITAL

# HCS EXAMPLE – ORGANIZATION STRUCTURE



# HCS EXAMPLE - PROCEDURE



# HCS EXAMPLE - TECHNOLOGIES

- PDA for doctors and nurses
  - Access to Hospital Data Base(DB), municipality DB
  - Medical forms compilation
  - Direct connections to other PDA
- GPS navigator for paramedics
  - Search fastest route



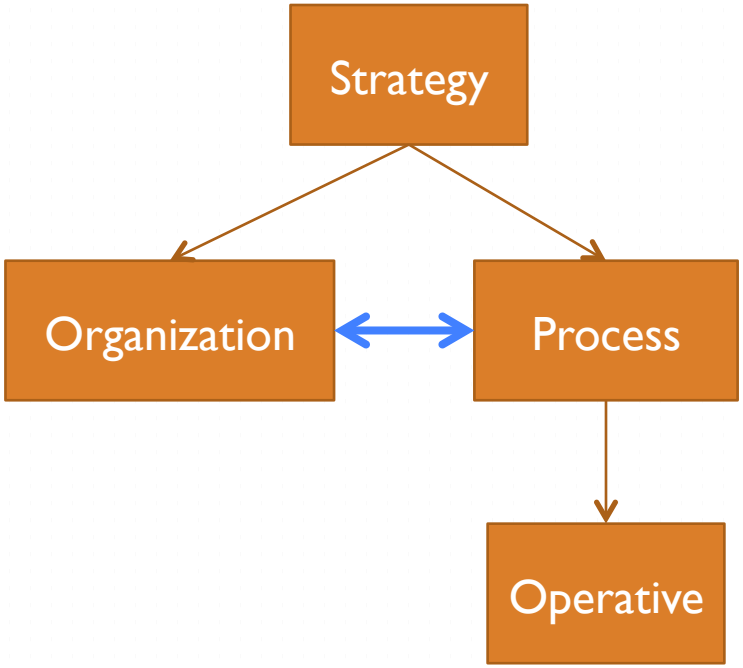


# LINKING STSs REQUIREMENTS

- Lot of languages characterize each type of STS requirements
  - Organizational structure requirements
    - E.g. UML deployment diagram[7], UML component diagram[7], MEMO[9], UEML[8]
  - Procedural requirements
    - E.g. BPMN[10], EPC[12], YAWL[11], UML sequence diagram[7], UML activity diagram[7]
  - Technological requirements
    - E.g. UML class diagram[7], Conspec [4], WSLA[13], WSAg[14]
- Are organizational, procedural and technological requirements independent?
  - Organizational and procedural requirements
  - Procedural and technological requirements

# BUSINESS MODEL CHARACTERIZATION[2]

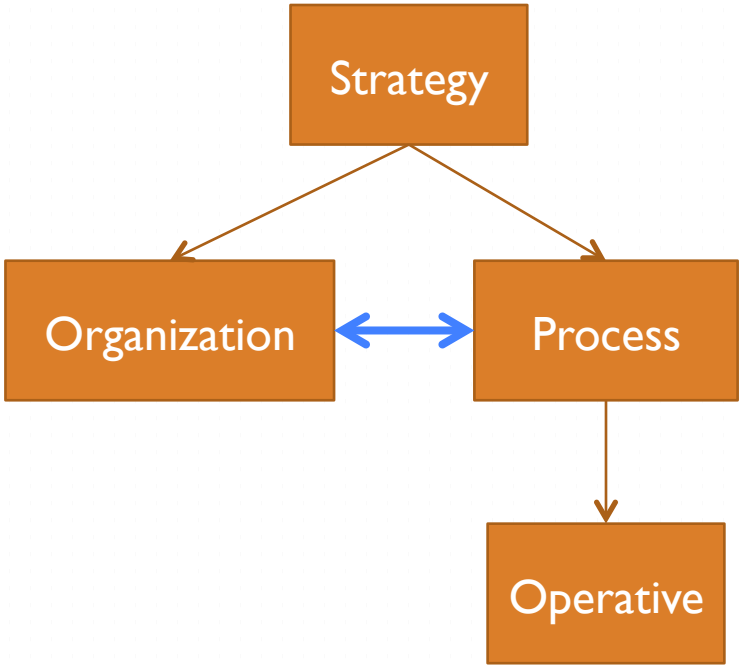
- Strategy
  - Long term objectives of a firm (5-10 years)
- Tactic
  - Medium term objectives of firm (2-5 years)
  - Organization
  - Process
- Operative
  - Short term objectives (up to 2 years)
  - Technologies used



# BUSINESS MODEL CHARACTERIZATION

Example:

- Strategy
  - Maintains high HCS surgery quality
- Tactic
  - Organization: at least two supervisors for every surgery session
  - Process: procedure to train new doctors
- Operative
  - All video of surgery sessions can be downloaded from an internal database



# SECURITY IN STSs

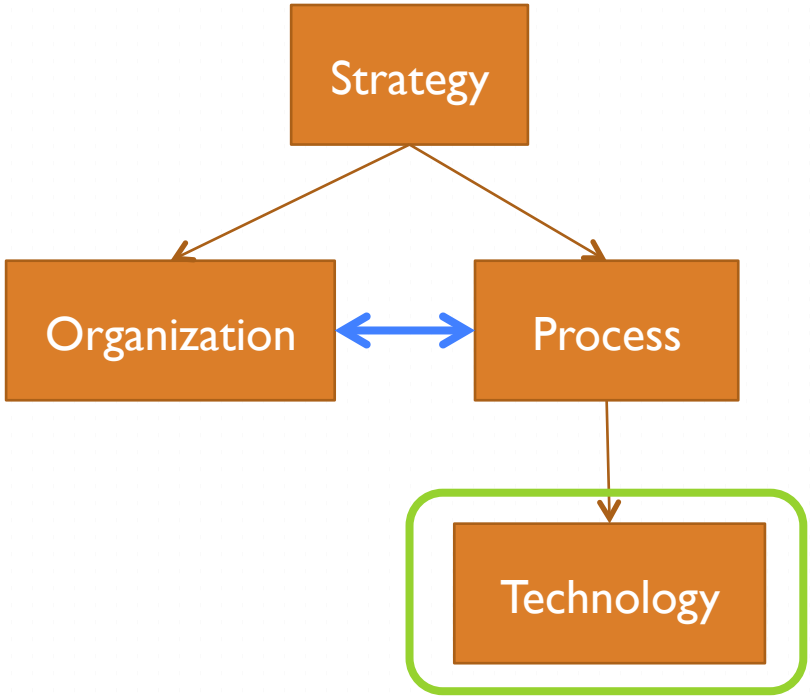
- Security is a central issue in STSs
  - Legal and financial consequences
- It is not only a technological problem
  - It is analyzed from as strategic, process, organizational and technological problems

# TECHNOLOGICAL SECURITY REQUIREMENTS

Security requirements about functional characteristics of STSs

Examples:

- Doctors' PDA transmits data using PGP encryption algorithm with 128 bits key length
- Paramedics' GPSs have a secondary battery

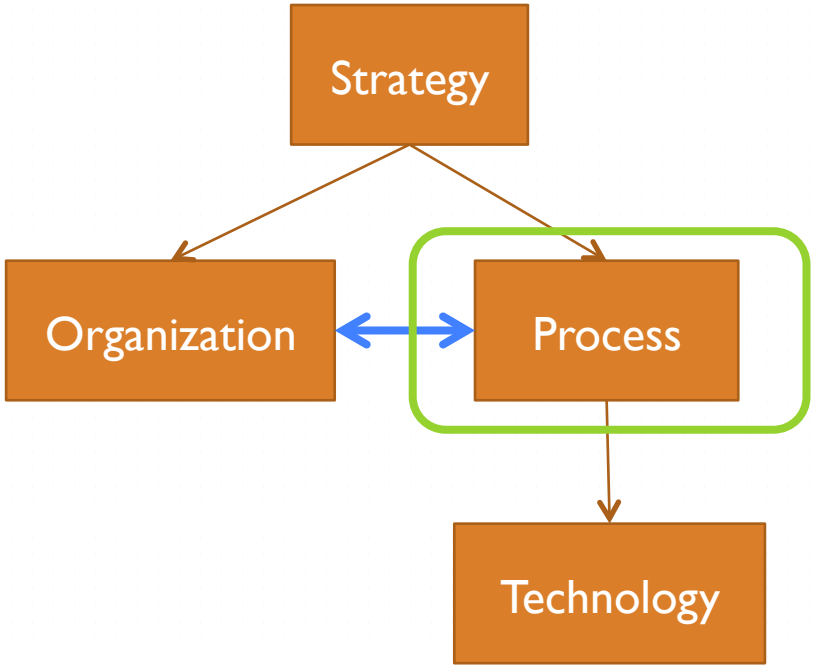


# PROCEDURAL SECURITY REQUIREMENTS

Security requirements about STS procedures

Examples:

- The HCS workflow manager receives and acknowledge from the performer who executes the activity "compute best route".
- The communication between activities "Compile medical forms" "Retrieve patient's info" is encrypted

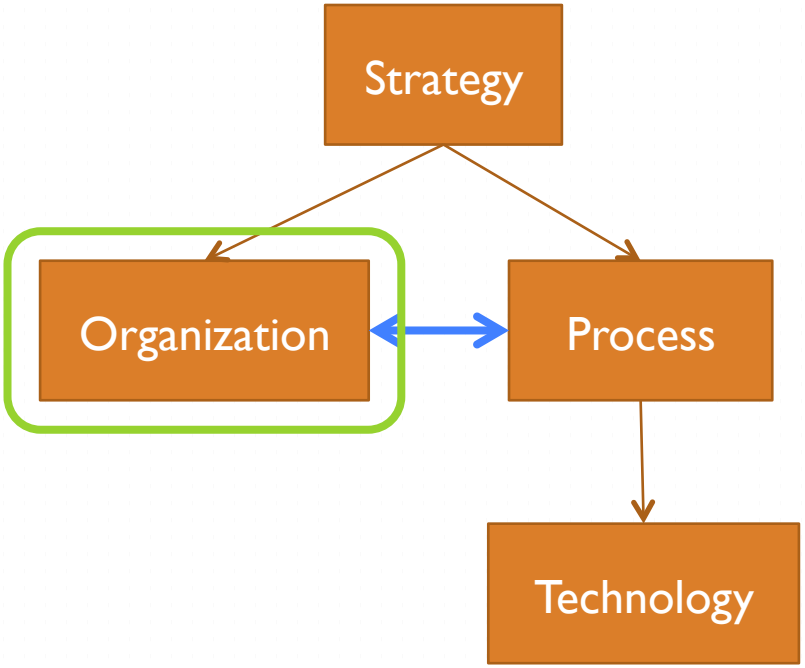


# ORGANIZATIONAL SECURITY REQUIREMENTS

Security requirements about organizational characteristics of STSs

Examples:

- Doctor will not repudiate the fact he/she healed a patient
- All HCS personnel will not disclose patients' sensitive information

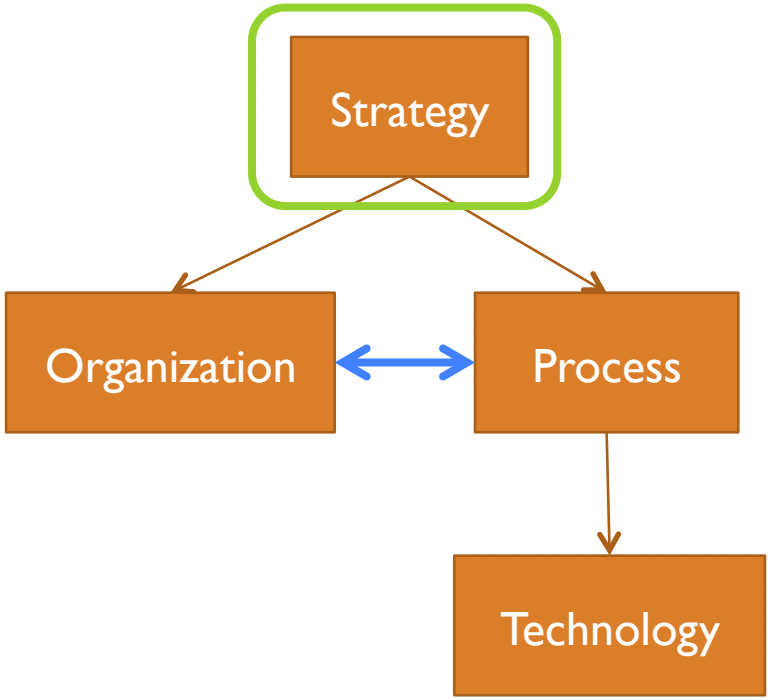


# STRATEGY SECURITY REQUIREMENTS

Security objectives linked to assets of STSs

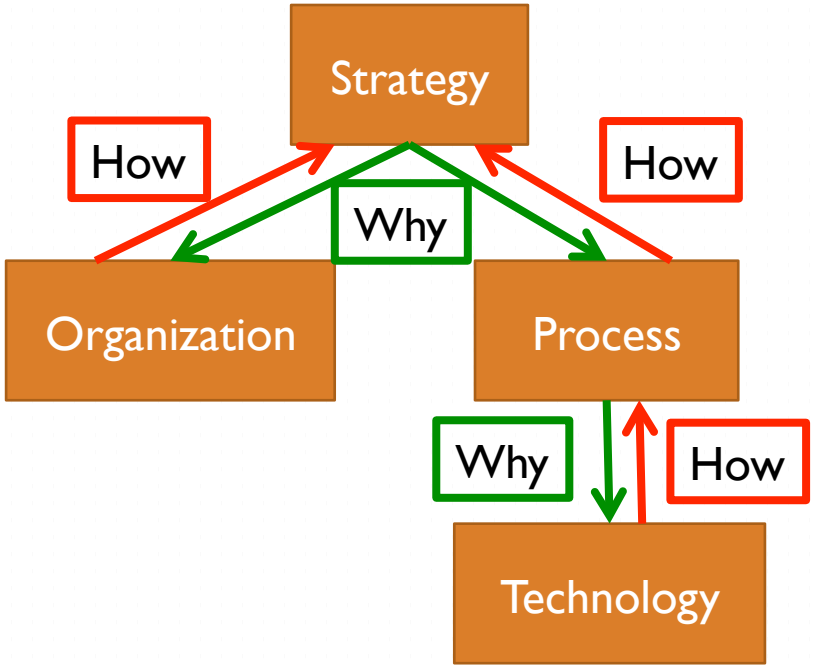
Examples:

- E.g. all HCS instruments will be used correctly
- E.g. reputation of hospital have to be preserved



# CONCEPTUAL LINKS

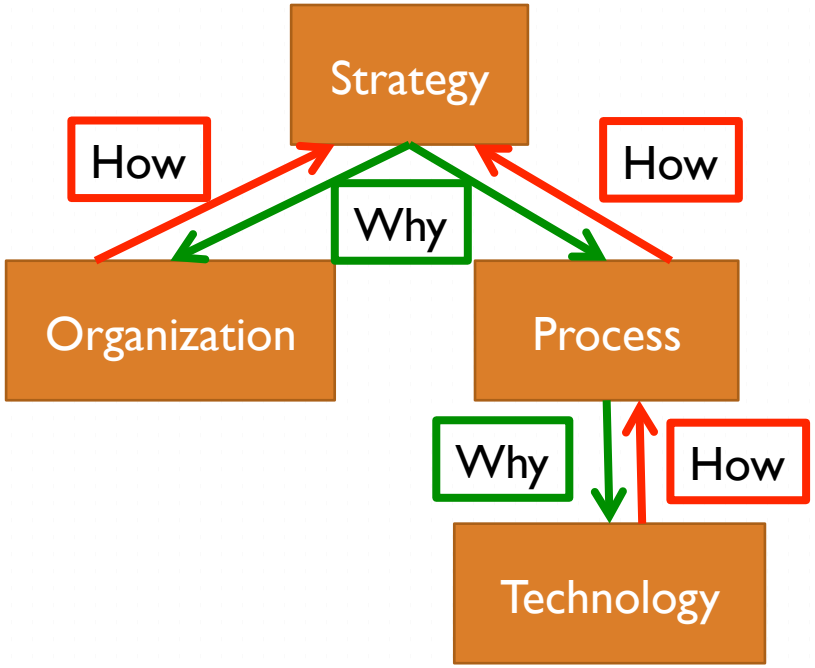
- Why/how relations
- Strategies answer questions on why a procedure/organization has certain characteristics
- Organizations/procedures answer questions on how a strategy is realized



# CONCEPTUAL LINKS

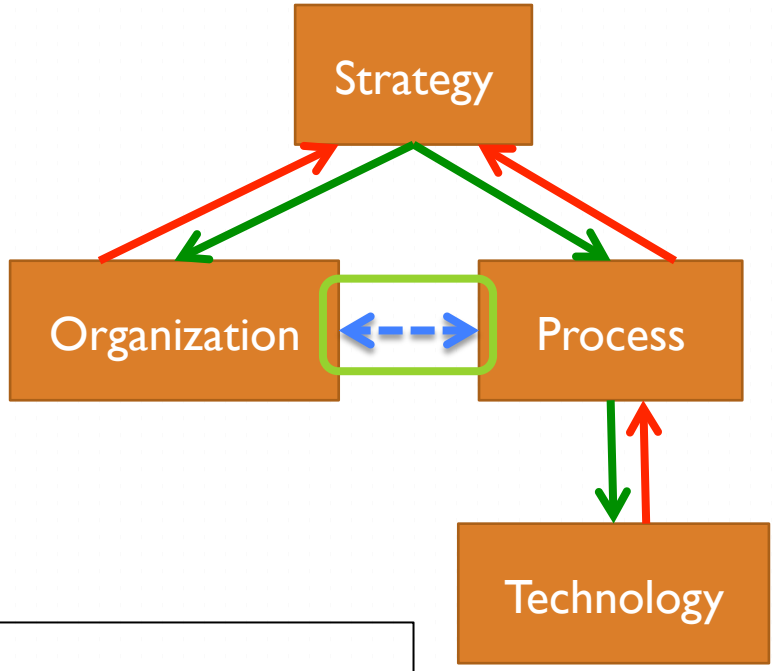
Example:

- Strategy
  - E.g. HCS sensitive information will not be misused
- Workflow
  - E.g. the communication between activities "Compile medical forms" and "Retrieve patients info" is encrypted



# ORGANIZATION AND WORKFLOWS LINK

- Organization and workflows security requirements must be coherent
- Process are designed on the basis of the organization structural requirements
- Procedural security requirements are directed influenced by organizational security requirements.



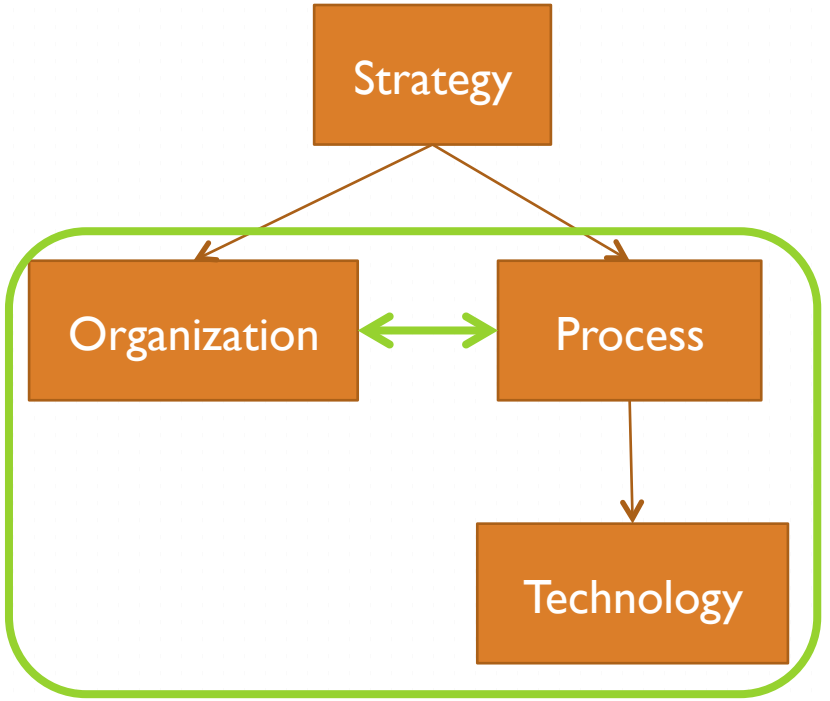

Example:

- Bind-of-duty between the roles of "ward responsible" and "nurse responsible"
- All the process where these two roles execute at least one activity, must enforce the bind-of-duty security requirements

# CHECKING ALIGNMENT OF SECURITY REQUIREMENTS

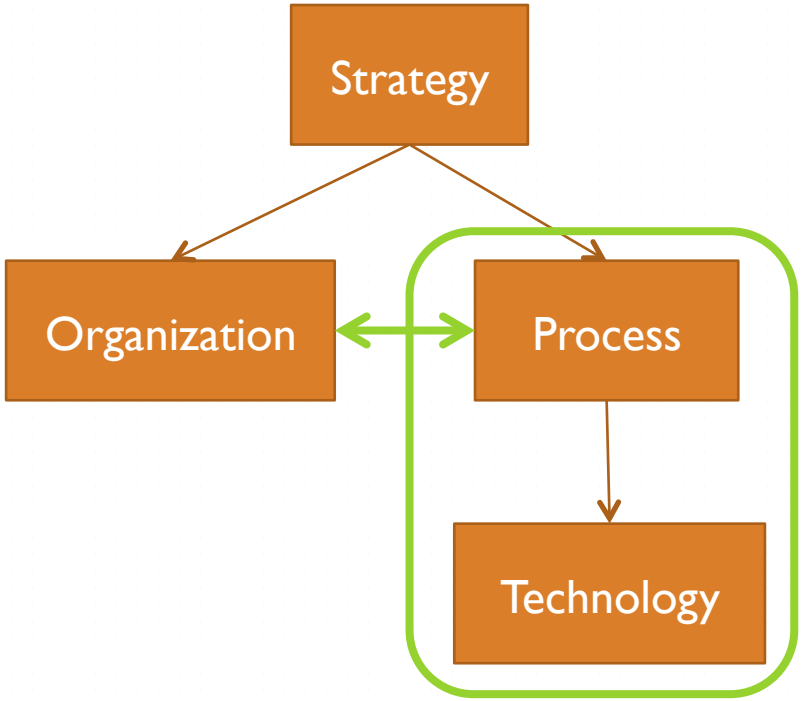
# ALIGNMENT OF SECURITY REQUIREMENTS

- Alignment: all security requirements are coherent
- If security requirements are not aligned consequences can be severe[3]

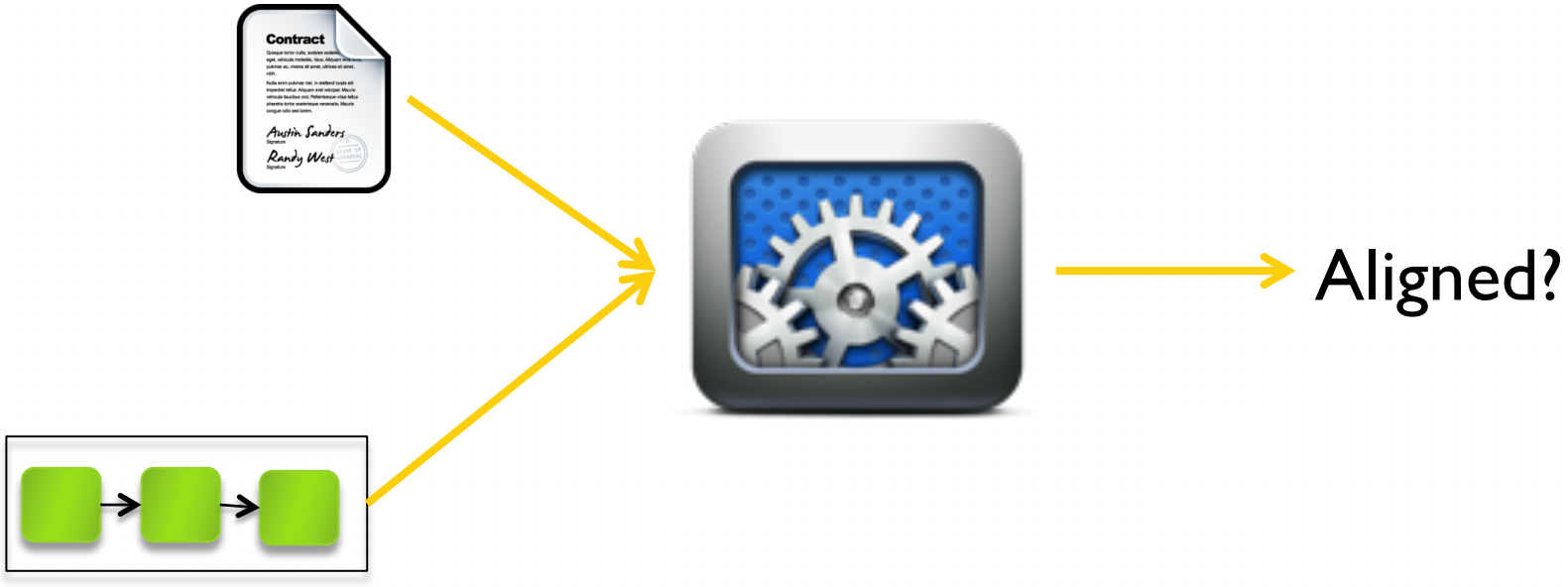


# PROCEDURAL-TECHNOLOGICAL ALIGNMENT

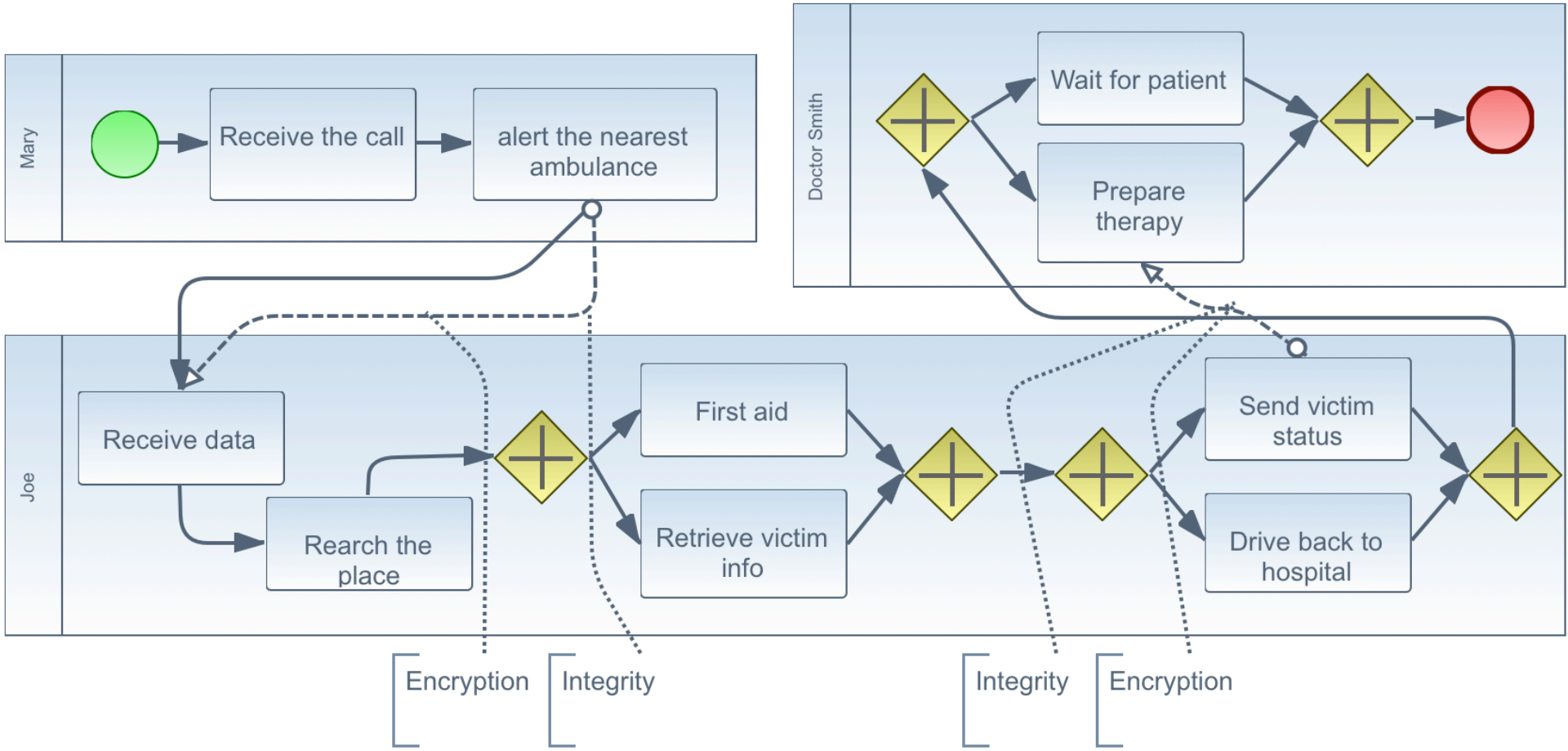
- Workflows security requirements
  - BPMN annotated with security requirements
- Technological security requirements
  - Conspec[4]
- Check
  - If security requirements expressed in annotated BPMN are enforced in Conspec.



# CHECKING ALIGNMENT ALGORITHM



# ANNOTATED BPMN 2.0

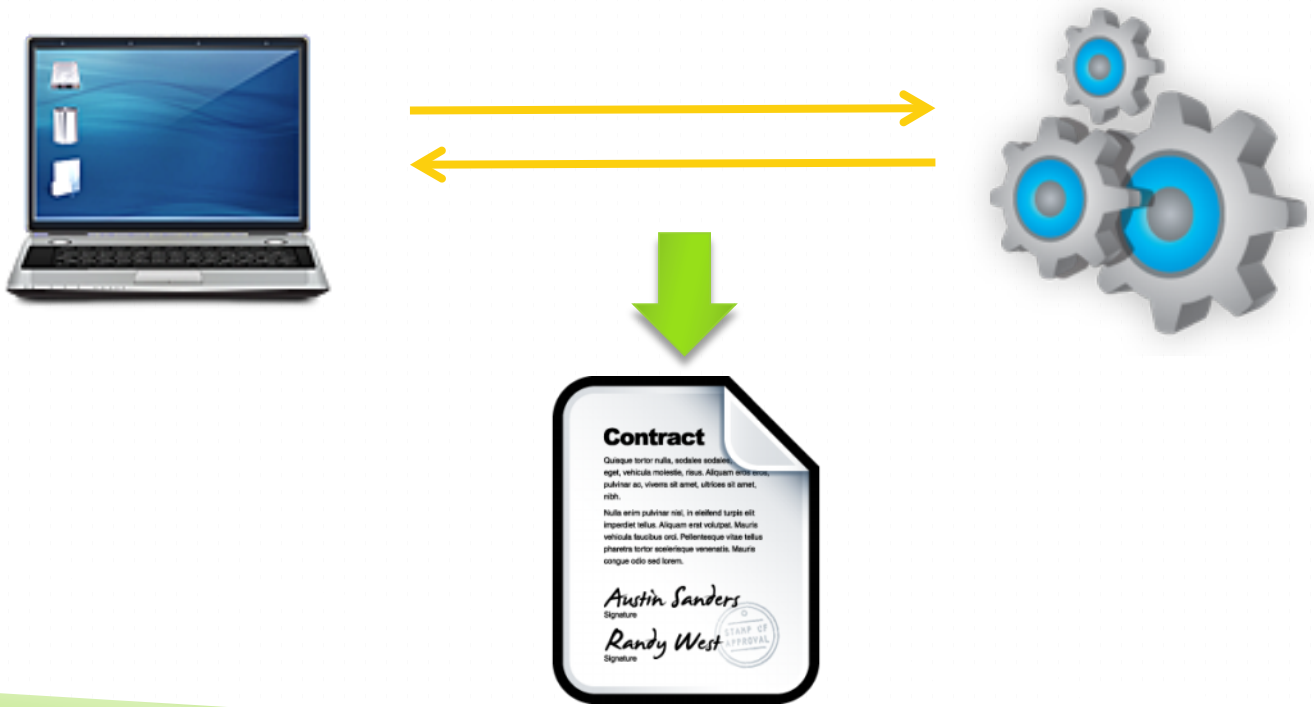


# PROCEDURAL SECURITY REQUIREMENTS

- **Integrity** (Call info, Alert the nearest ambulance, Receive data)
- **Encryption** (Call info, Alert the nearest ambulance, Receive data)
- **Integrity** (Victim status, Send victim status, Prepare therapy)
- **Encryption** (Victim status, Send victim status, Prepare therapy)

# CONTRACTS

- STSs : interaction between autonomous technological components
  - Service-oriented Architecture (SOA)



# CONSPEC

SECURITY STATE

string sendingService = <S1>;

string receivingService = <S2>;

string CommunicationName = <C1>;

Boolean encrypt = false;

**BEFORE** v#activity.end(string name, int time, int date, string exec, stream output) **PERFORM**
name == sendingService && encrypted == v#input.isEncrypted(PGP)—> {skip}

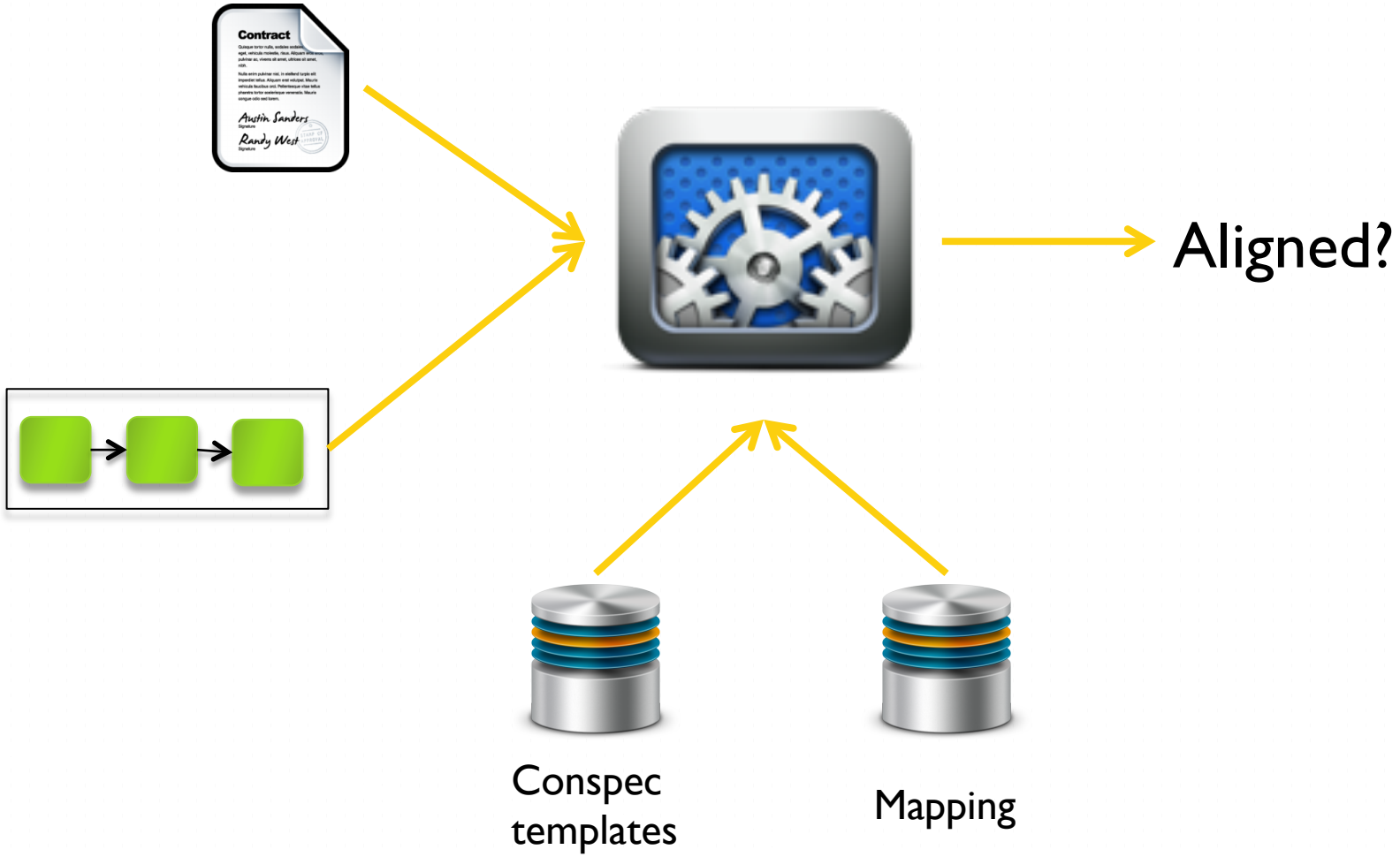
!(name == sendingService) —> {skip}

**BEFORE** v#activity.start(string name, int time, int date, string exec, string input) **PERFORM**

name == receivingService && communicationName==input.getName && encrypted(PGP) —> {skip}
name == receivingService && !communicationName==input.getName —> {skip}

!(name == receivingService) —> {skip}

# CHECKING ALIGNMENT ALGORITHM



# STEPS FOR CHECKING ALIGNMENT

1. Translate parameters of procedural security requirements
2. Instantiate procedural security requirements
3. Search technological security requirements

# 1- TRANSLATION OF SECURITY PROPERTY PARAMETERS

- Change parameters using a map provided by user

**Example:**
**Encryption** (Call info, Alert the nearest ambulance, Reach the place)

| BPMN element | Conspec element |
|---|---|
| Call info | healthcare.gov/res/callInfo |
| Alert the nearest ambulance | healthcare.gov/services/alertParamedic |
| Receive data | healthcare.gov/services/ambulance/receiveData |

**Encryption** (healtcare.gov/res/callInfo, healtcare.gov/services/alertParamedic, healtcare.gov/services/receiveData)

# 2- INSTANTIATE PROCEDURAL SECURITY REQUIREMENTS

- Generate the Conspec contract correspondent to the security policy
  - **Encryption** (healtcare.gov/res/callInfo, healtcare.gov/services/alertParamedic, healtcare.gov/services/receiveData)
- Use repository of Conspec contract template
  - Conspec contracts with placeholders

# CONSPEC - TEMPLATE

SECURITY STATE

string sendingService = **<S1>**;

string receivingService = **<S2>**;

string CommunicationName = **<C1>**;

Boolean encrypt = false;

**BEFORE** v#activity.end(string name, int time, int date, string exec, stream output) **PERFORM**
name == sendingService && encrypted == v#input.isEncrypted(PGP)—> {skip}

!(name == sendingService) —> {skip}

**BEFORE** v#activity.start(string name, int time, int date, string exec, string input) **PERFORM**

name == receivingService && communicationName==input.getName && encrypted(PGP) —> {skip}
name == receivingService && !communicationName==input.getName —> {skip}

!(name == receivingService) —> {skip}

# 3- SEARCH TECHNOLOGICAL SECURITY REQUIREMENTS

- Search the generated policies in all contracts
- Text search :
  - Bitap [5]
    - approximate string matching algorithm O(nm)*
  - Boyer et al. [6]
    - most efficient string search algorithm O(n)*

* **m** is the length of the pattern and, **n** is the length of the searchable text.

# CHECKING COMPLIANCE ALGORITHM

```
1  checkAlignmentSO (
2  Set<SecurityProperty> SP_set, Set<Template> template_set,
3  Set<Contract> contract_set, Mapping mapping
4  )
5
6  Boolean aligned = true
7  while !SP_set.empty && aligned
8    SecurityPropery SP = SP_set.pop
9    Set<TranslatedSecurityPropery> TSP_set=translate(SP, mapping)
10   Set<InstantiatedSecurityPolicy> ISP_set=instantiate(TSP_set, template_set)
11   for each  Contract C in contract_set
12     if ISP.involvedParty.contains(contract.provider)
13       Set<Contract> involvedContract_set.insert(C)
14   Boolean found = false
15   while !found && aligned
16     InstantiatedSecurityPolicy ISP = ISP_set.pop
17     for each contract C in involvedContract
18       if search(ISP, C)
19         found = true
20         break
21     if found = = false
22       aligned = false
23       break;
24 return aligned
```

# SUMMARY – FUTURE WORK

- We established link between different type of security requirements
  - Formalize link
- We defined the algorithm which checks the alignment between procedural and technological security requirements
  - Implement the algorithm
- We defined a semi-automated framework which checks alignment of organizational, procedural and technological security requirements.
  - Validate the framework with case studies

# THANK YOU!

# REFERENCES


1. http://www.businessdictionary.com/definition/organizational-structure.html
2. Alexander Osterwalder. **The Business Model Ontology - a proposition in a design science approach.** 2004
3. R. Crook, D. Ince, L. Lin, and B. Nuseibeh. Security requirements engineering: **When anti-requirements hit the fan.** In Proc. of RE'02, pages 203–205. IEEE, 2002.
4. Irem Aktug and Katsiaryna Naliuka. **Conspec - a formal language for policy spec- ification.** Electronic Notes in Theoretical Computer Science, 197(1):45 – 58, 2008. Proceedings of the First International Workshop on Run Time Enforcement for Mobile and Distributed Systems (REM 2007).Ricardo A. Baeza-Yates and Gaston H. Gonnet. **A new approach to text searching.** Commun. ACM, 35(10):74–82, 1992.
5. Robert S. Boyer and J. Strother Moore. **A fast string searching algorithm.** Commun. ACM, 20(10):762–772, October 1977.
6. Michael Hammer and James Champy (1993). **Reengineering the Corporation: A Manifesto for Business Revolution,** Harper Business

# REFERENCES


7. http://www.uml.org
8. F.Vernadat (2002): **UEML: Towards a unified enterprise modelling language**, International Journal of Production Research, 40:17, 4309-4321
9. Frank, U., "**Multi-perspective enterprise modeling (MEMO) conceptual framework and modeling languages**," System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on , vol., no., pp.1258,1267, 7-10 Jan. 2002
10. P.Wohed, W.M.P.Aalst, M. Dumas, A.H.M. Hofstede, and N. Russell. **On the suitability of bpmn for business process modelling.** Business Process Management, volume 4102 of Lecture Notes in Computer Science, pages 161–176. Springer Berlin Heidelberg, 2006.
11. W. M. P. van der Aalst and A. H. M. ter Hofstede. Yawl: yet another workflow language. Inf. Syst., 30(4):245–275, June 2005.
12. W.M.P. van der Aalst. **Formalization and verification of event-driven process chains.** Information and Software Technology, 41(10):639 – 650, 1999.

# REFERENCES


13. http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf
14. http://www.ogf.org/documents/GFD.107.pdf