

Università di Camerino
Dipartimento di Matematica e Informatica

Designing Modular Agent Systems

Diego Bonura
Dip. di Matematica e Informatica
Università di Camerino
Camerino, Italy
diego.bonura@unicam.it

Leonardo Mariani
Dip. di Informatica, Sistemistica e Comunicazione
Università di Milano Bicocca
Milano, Italy
mariani@disco.unimib.it

Emanuela Merelli
Dip. di Matematica e Informatica
Università di Camerino
Camerino, Italy
emanuela.merelli@unicam.it

Motivations

- The agent technology has been used in several application domains, such as:
  - Manufacturing
  - e-commerce
  - Information Retrieval

  [Shen et al.99,Klusch01,Guttman98] for surveys

- Different agent-based systems for different application domains may have several similarities
  - communication paradigms, yellow page services, mobility, services provisioning ...

Modularity of the system and **binary code reuse** reduce efforts in developing new systems

Our Approach

- We developed new multi-agent systems (MASs) based on a layered component-based software architecture
  - layers to master complexity
  - components to strength modularity and binary code reuse

- We experimentally evaluated the approach on
  - the biological domain
  - the manufacturing domain

In the following, we present our practical experience in two steps

1. by introducing the design of the MAS for a biological domain,
2. by analyzing issues on reusing components of the biological domain into the manufacturing domain

.

Biological domain 1/2

- The amount of information is daily increasing (NCBI, GenBank, SwissProt, SRS, Ensembl, …)
- Information are wide distributed
- Information are heterogeneous
- Users are tied to directly manage the activities of their experiments
  - time-consuming
  - manually accessing and manipulating information
- Users need to run own proprietary algorithms on remote data site

Biological domain 2/2

Requirements: Retrieval System capable of

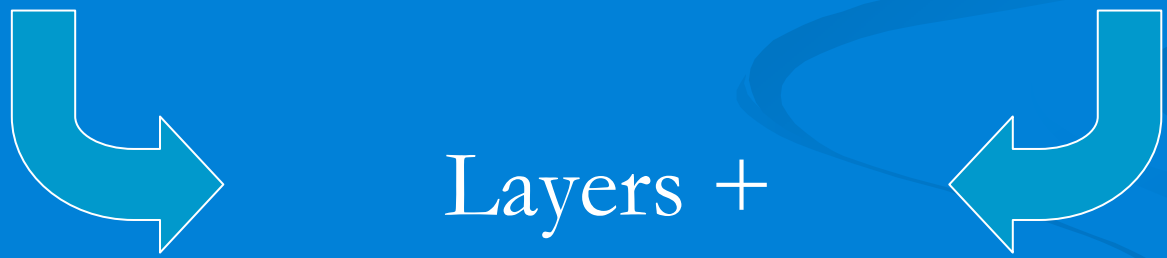
- specifying complex selection criteria
- accessing to heterogeneous sources (HTML, FASTA,…)
- supporting the specification and the execution of workflows of activities

Our solution: MAS with following features

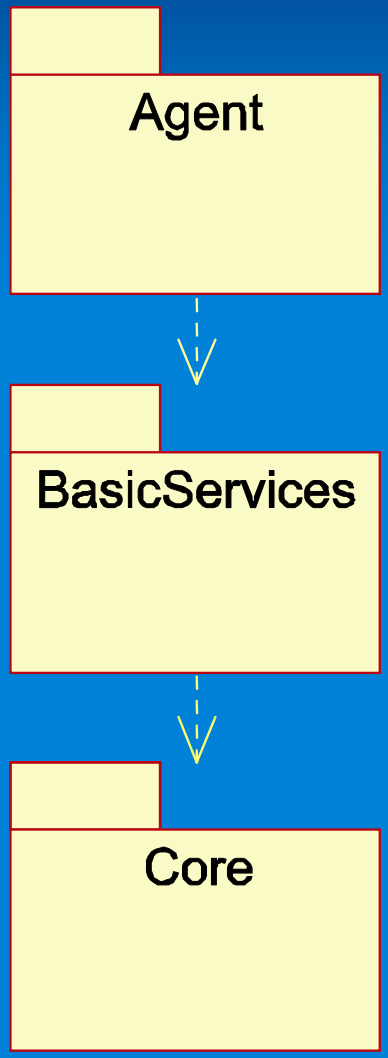
- Security, Mobility, …

System Architecture Style

- Layers
  - Master complexity
  - Facilitate security
- Components
  - Enhance flexibility
  - Enhance reuse



System Design – Three-layered architecture

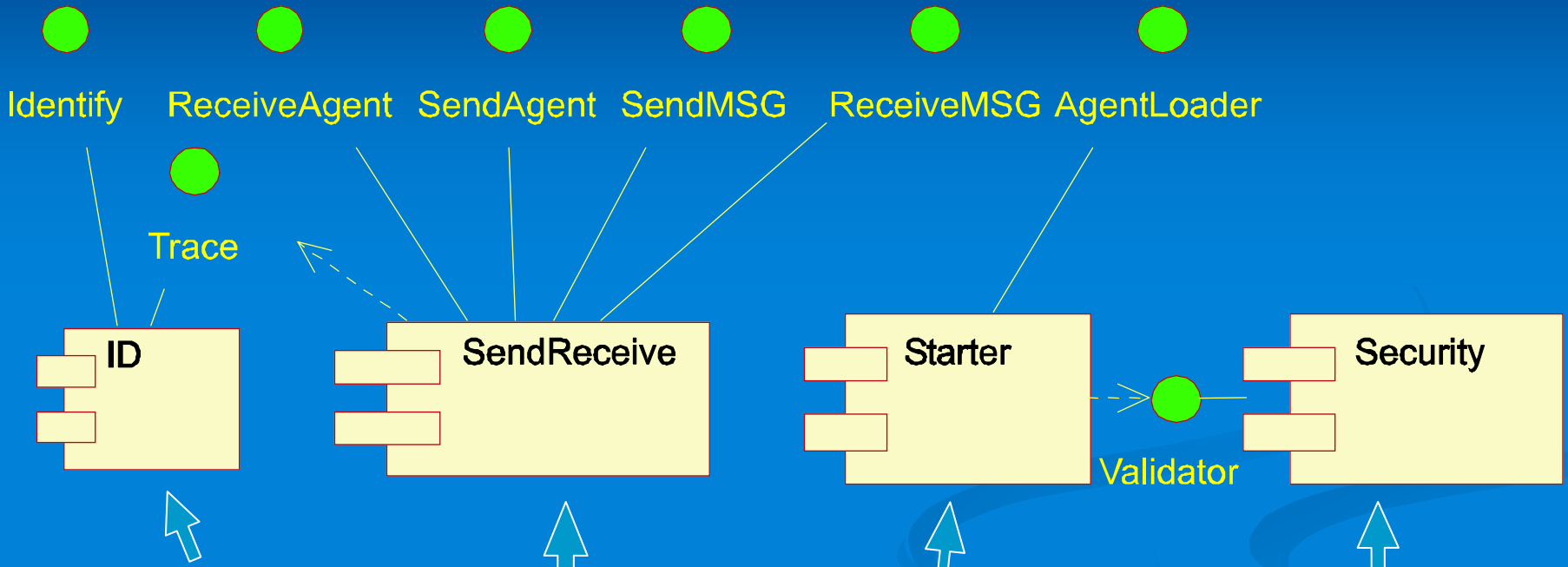


- User agents (biological agents)
- Application service agents

- System services to supports agent activities
  - mobility,
  - agent communication protocols
  - security policies …

- Primitives similar to those in an O.S kernel
  - communication protocols,
  - traceability,
  - security

System Design – Core Layer



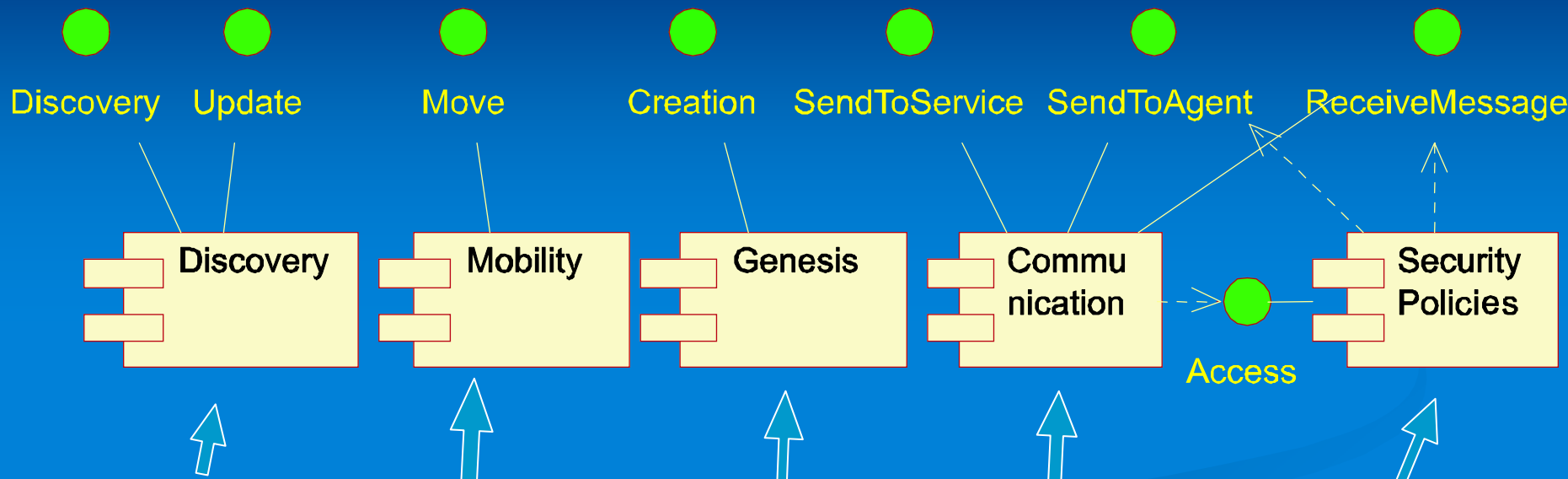
- Implements identity management functions
- Manages repositories

- Executes inactive agent (created or migrated)

- Implements low level inter-platform communication

- Checks Java code searching for malicious agents

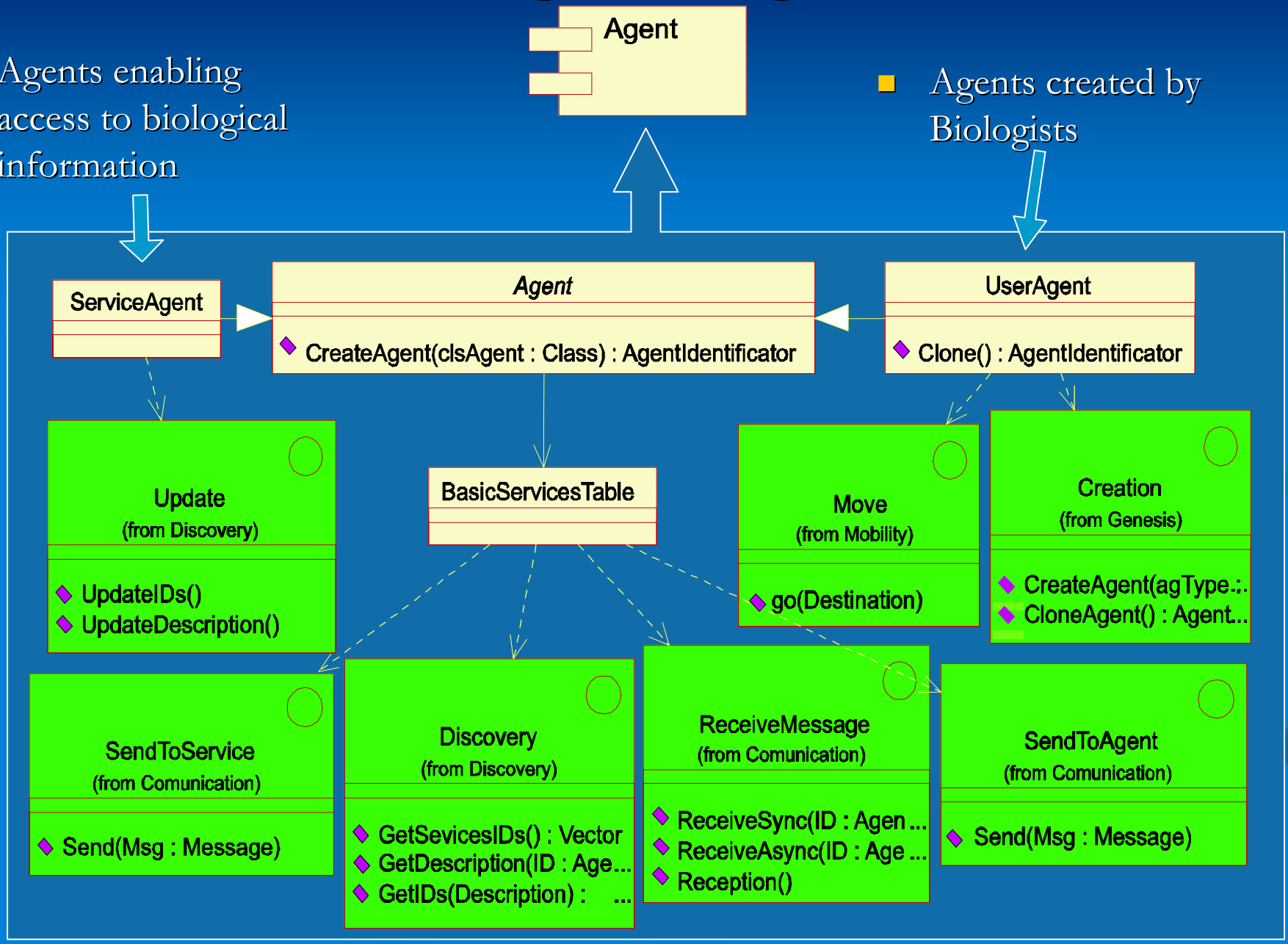
System Design – Basic Service Layer



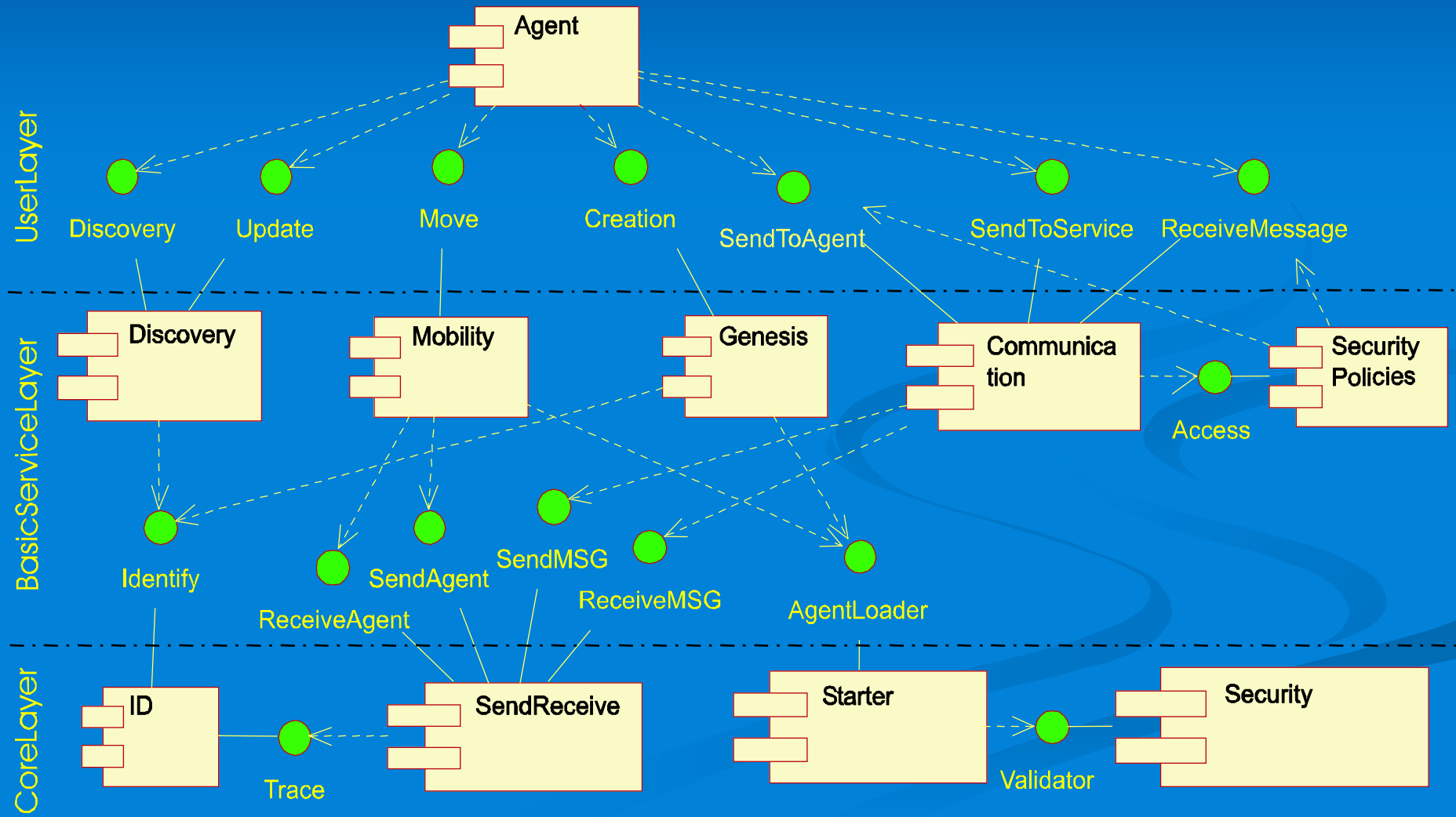
- Detects local service agents
- Enables code mobility across platforms
- Implements agents creation (cloning)
- Implements agents communication by intra and inter-platform messages
- Protects services from user agents

System Design – Agent Layer

- Agents enabling access to biological information
- Agents created by Biologists



System Design – Final view

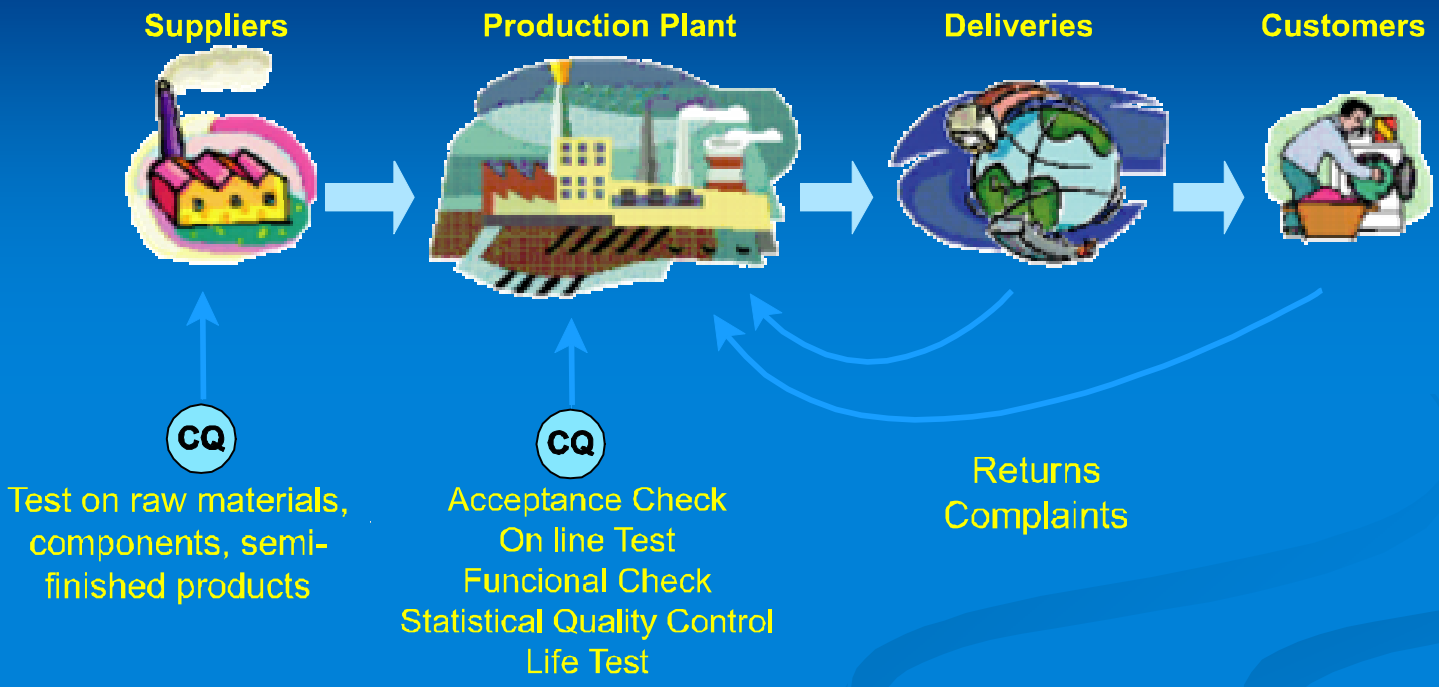


System Design – remarks

- Explicit description of dependencies among components
  - Dependencies analysis techniques
  - Possibility to measure coupling
- Refactoring
  - white-box components enable refactoring of both components arrangement and components internal structure
- Three levels of abstraction
  - Core Layer: general features
  - BasicService Layer: common features related to the requirements of the application domain
  - User Layer: features of the application domain

from Biological to Manufacturing Domain: the MAS Design

Manufacturing domain 1/2



- The production processes are performed by both sequential and parallel activities
- A complex set of activities defines complex production paths
- Each single supplier uses his own quality control mechanism
- Quality reports are distributed and heterogeneous

Manufacturing domain 2/2

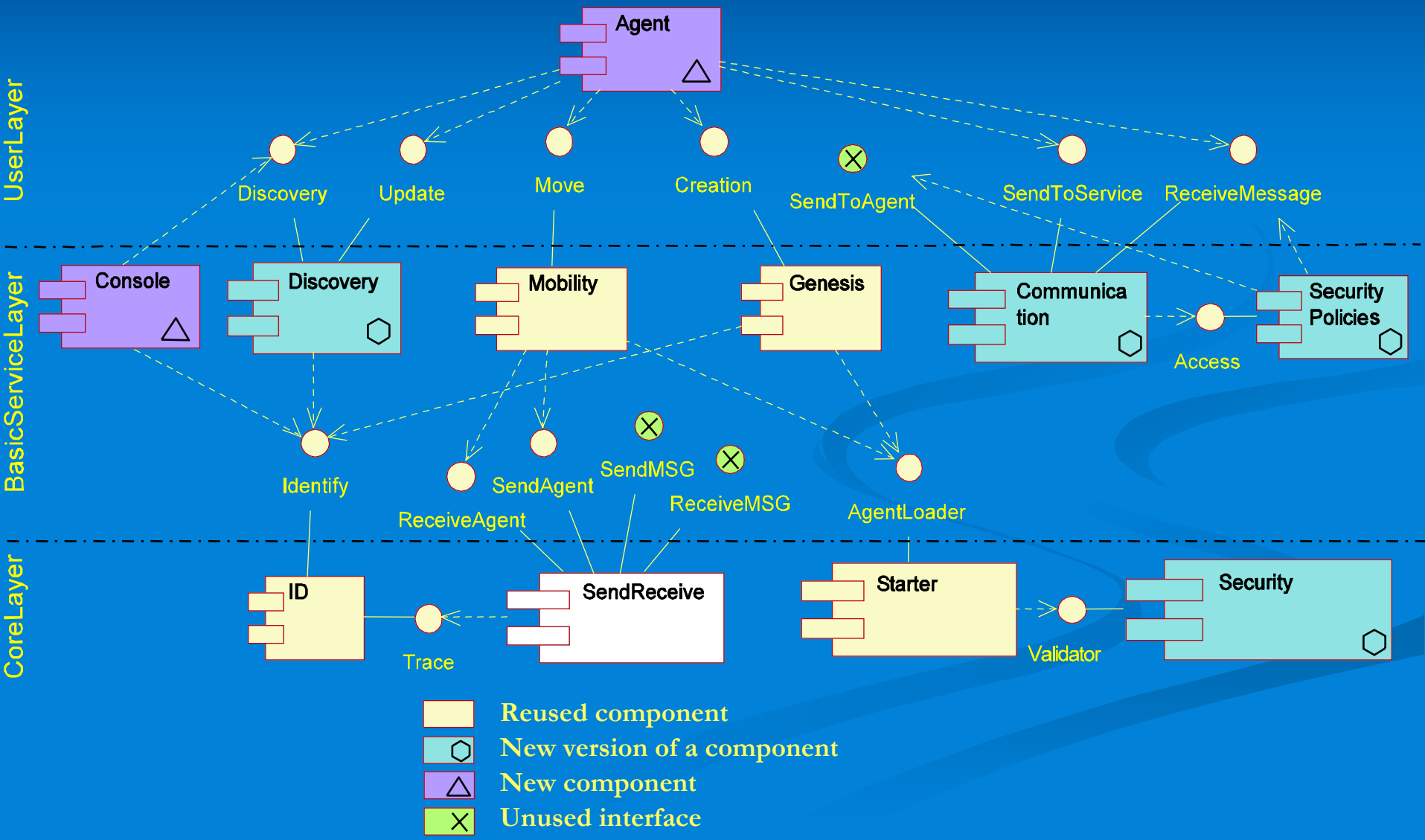
Requirements: system capable of

- tracing lifecycle of both components and semi-manufactured products to discover executed tests and existing quality reports
- accessing to heterogeneous sources (different embedded control systems)
- integrating information from quality reports of different manufacturers

Our solution: MAS with following features

- Security, Monitoring, Mobility, …

Manufacturing domain – System Design



Manufacturing domain
remarks on component reuse

- Components can be effectively reused in different domains
  - better in lower layers than in higher layers,
  - in fact, only two components were developed from scratch; they belong to the 2nd and 3rd layers respectively
- Reusing components allows also reusing
  - knowledge,
  - specification,
  - documentation,
  - test cases, ...

Conclusions

- The components plus layers architectural style is not applicable on any system, but it seems to be suitable for agent-based systems
- The components plus layers architecture style exploits several benefits
  - a modular and disciplined design phase
  - a functionality can easily be addressed at the different levels of abstraction
- Reuse remarks
  - components located in lower layers are very likely reusable, but components located at higher layers of the system are hardly likely reusable