

Autonomous services

Feedback from some on-going EU research projects

Françoise Baude

OASIS research group at Univ. Nice Sophia-Antipolis, CNRS-I3S, INRIA Sophia-Antipolis





Objectives

- Give a synthetic (modest and personal) view on on-going efforts w.r.t. Autonomic (application level) Services:
 - Requirements and needs wrt to
 - Software architecture
 - Programming efforts
 - How solutions currently defined in ...
 - Unit GRID funded projects like CoreGRID, GridComp
 - Unit FET “Autonomic Situated Communications” project like BIONETS
 - ... try to address those requirements, which are somehow common
- Is a convergence in progress ?
- Are there opportunities for stronger exchanges, mutual collaborations ?



Autonomicity: from adaptivity to evolution

SASO 2007



First IEEE International Conference on Self-Adaptive and Self-Organizing Systems
Boston, Mass., USA, July 9-11, 2007

Goal: laying the foundations of a new principled approach to engineering systems ... that are

- **Self-adaptive** systems work in a top-down manner. They evaluate their own global behavior and change it when the evaluation indicates that they are not accomplishing what they were intended to do, or when better functionality or performance is possible. Such systems typically operate with an explicit internal representation of themselves and their global goals.
- **Self-organizing** systems work bottom-up. They are composed of a large number of components that interact according to simple and local rules. The global behavior of the system emerges from these local interactions, and it is difficult to deduce properties of the global system by studying only the local properties of its parts. Such systems do not use internal representations of global properties or goals; they are often inspired by biological or sociological phenomena.

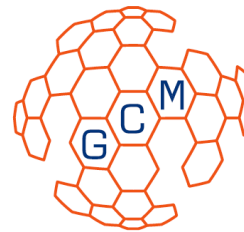
Autonomic Computing Project Repository and Events <http://autonomiccomputing.org>

Autonomic Grid Applications

- Feasibility of automatic adaptation of
 - structured parallel programs : ASSIST  
- Currently being extended and generalized with the context of the Grid Component Model (GCM)

Core **GRID** 

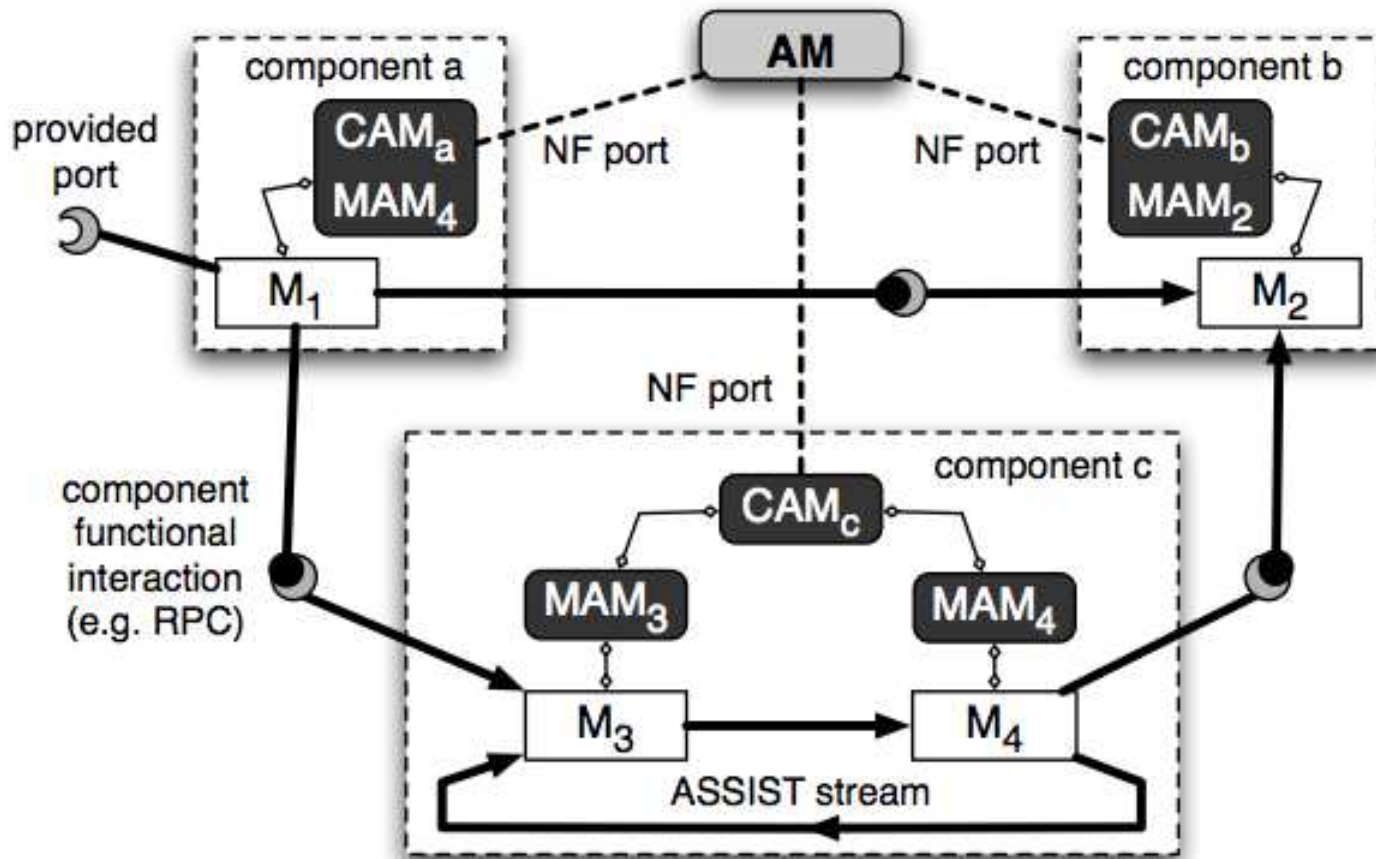
GridCOMP
Effective Components for the Grids



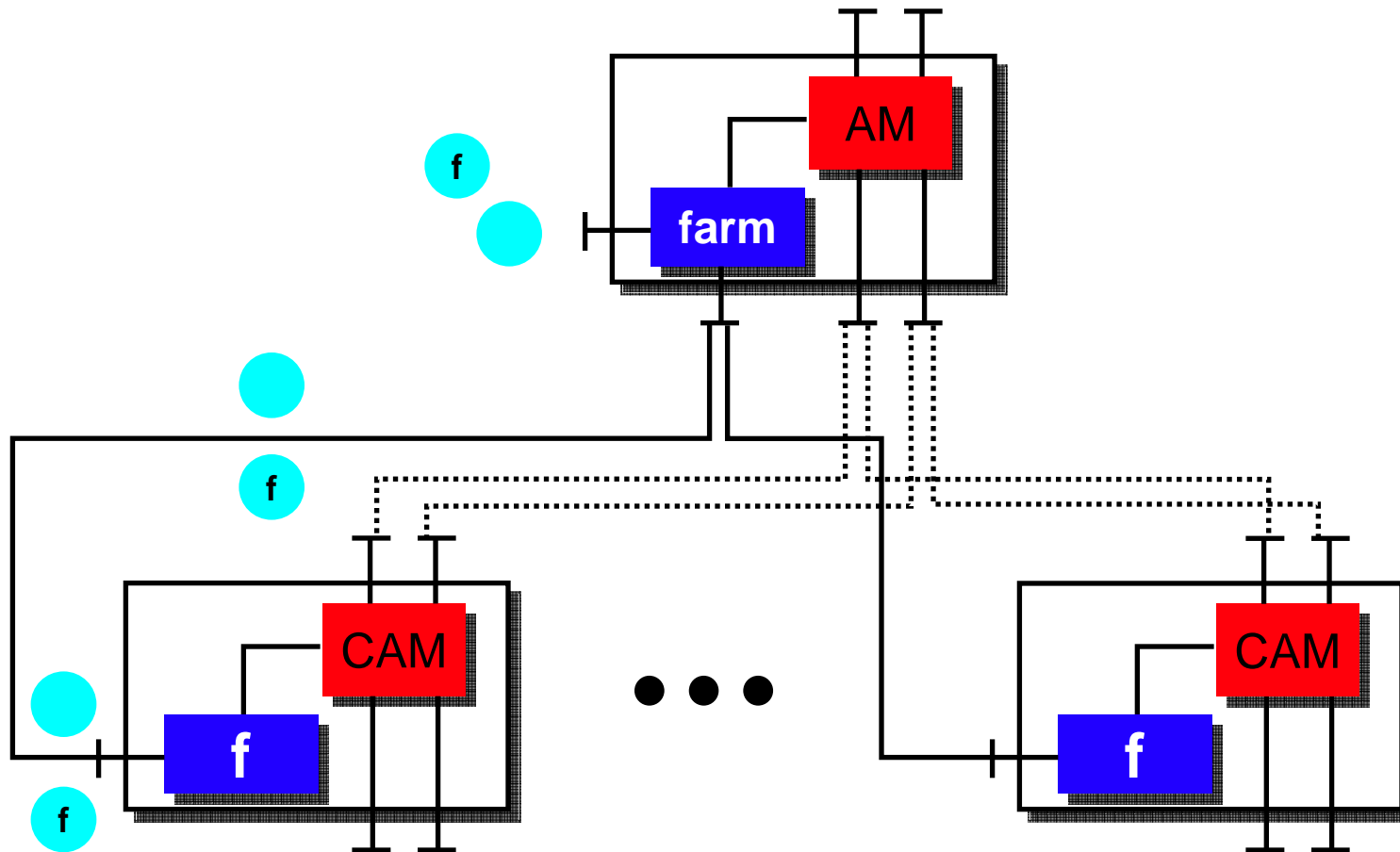
ASSIST super components

- generic graph of components (pipeline as a subcase) & component task farm
- with managers (module, component and application manager);
 - non functional part of the components, takes care of ensuring a performance contract
 - user defined performance contract, best effort policy
 - module manager (MAM) ⇒ parmods,
component manager (CAM) ⇒ (composite) components,
application managers (AM) ⇒ complete application

ASSIST manager hierarchy



ASSIST supercomponents hierarchy

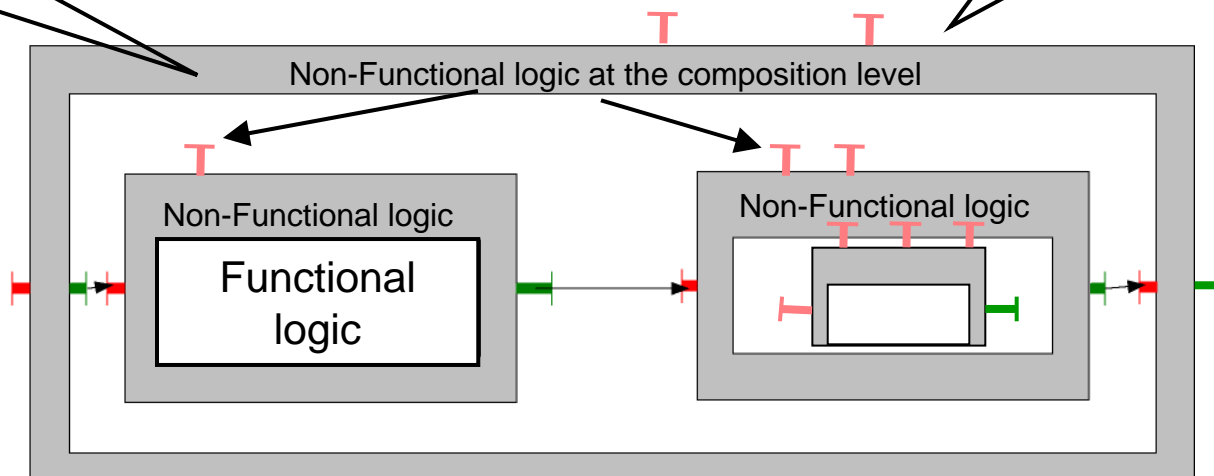


Fractal Composite components

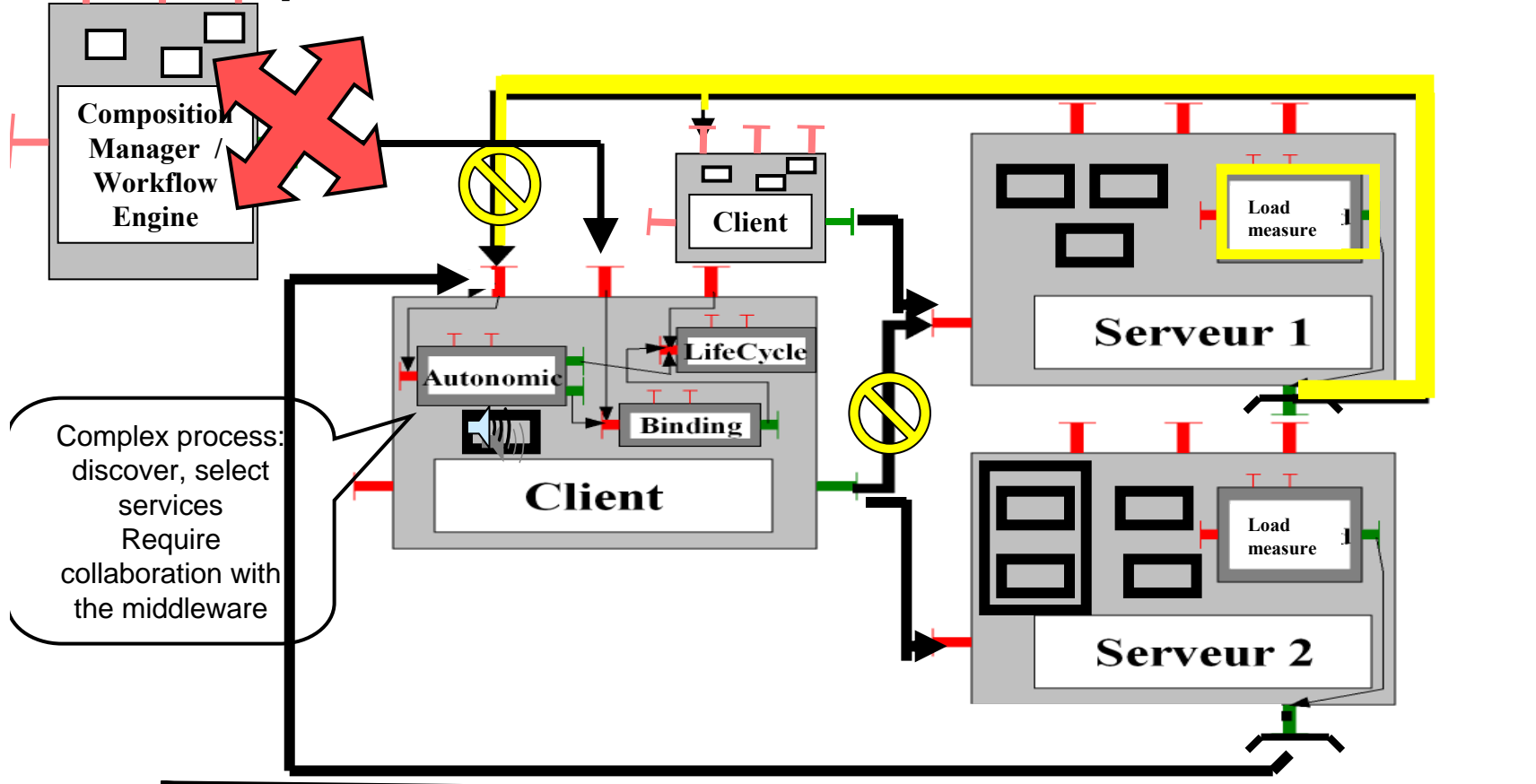
Delegated and Hierarchical composition of the non-functional logic of the component

```
<definition name="Client-Server">  
<interface name="s" role="server" signature="java.lang.Runnable"/>  
<interface name="c" role="client" signature="MyService"/>  
<component definition="ClientDefinition" name="client" >  
<component definition="ServerDefinition" name="server" >  
<binding client="this.s" server="client.s"/>  
<binding client="Client.c" server="server.s"/>  
<binding client="Server.c" server="this.c"/>  
</definition>
```

Self-similarity



GCM components: towards autonomic composite services



GCM preliminary status: support for a wide panel of composition solutions (as static / dynamic / autonomic) looks feasible



Self-managing applications on the Grid

[AutoMate, M. Parashar @ Rutgers U.]

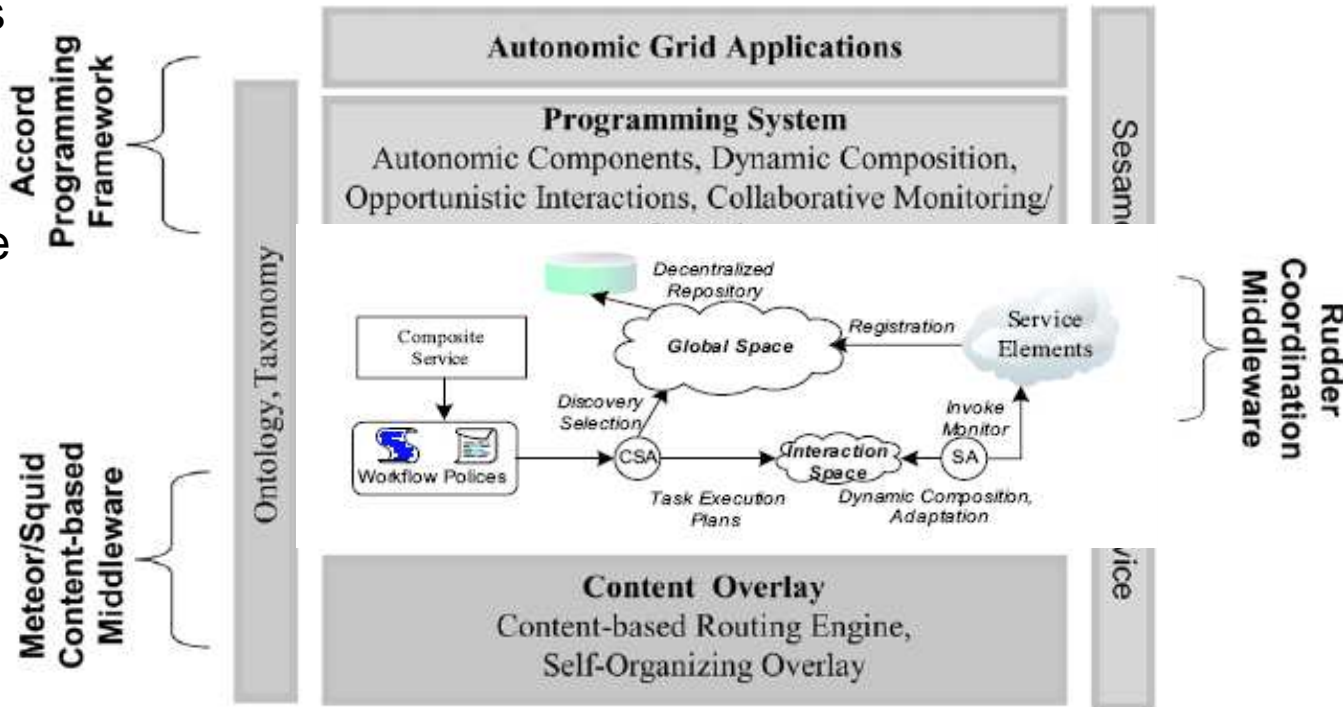
- Constraints from (large-scale) Grid (Global computing) environments and applications
 - Dynamicity and failures
 - Incomplete knowledge of the global system state
- Requirements behind a self-management feature
 - Static application requirements and system and application behaviours to be relaxed
 - Behaviours of elements and applications to be sensitive to the dynamic state of the system and adapt to these changes at runtime
 - Required common knowledge be expressed semantically rather than in terms of names, addresses and identifiers
 - Core enabling middleware services be driven by such a semantic knowledge
- Solution defined in the AutoMate project
 - Grid application: a set of self-managing components, driven by rules
 - Clear separation between functional, sensor/actuators, rules injection ports

AutoMate

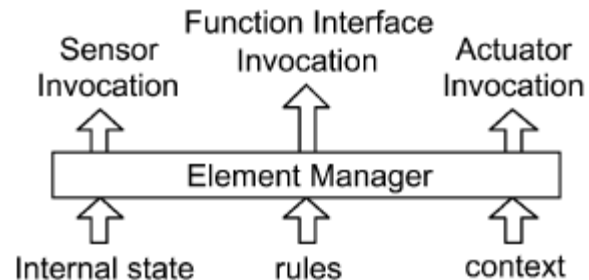
Composition is translated into interaction rules injected into CEs by the Composition Manager

These rules form the basis of a truly autonomic composition mechanism : Rudder (based on external agents, not the CEs)

ACCORD decouples interaction and coordination from computation, and let them be managed via rules



Element Manager	Autonomic Element
Computational Element	Operational Port
	Control Port
	Functional Port





Autonomic Composition of Services

- A concern in numerous research communities
 - Grid computing
 - Software and Services
 - Networked Communications
 - ...
- From static to dynamic, up to autonomic composition of services
 - Keep the user view simple, so transfer the complexity to the composition engine, the middleware, the system...
 - Flexible and late bindings (flexible aggregation)
 - Resolution of interface mismatches
 - Constraint-Satisfaction Problem, Genetic, service's graph abstract-concrete mapping, ..., for service selection

BIONETS: Don't Reinvent Nature, Just Go For It

Daniele Miorandi
CREATE-NET

LEARNING FROM EXAMPLES

- ▶ Pervasive environments will present scale and complexity figures not far from those typical of biological/socio-economical systems
- ▶ These 3 issues (heterogeneity, scalability, complexity) already successfully tackled by *Nature & Society*
- ▶ Plenty of examples of biological/socio-economical systems able to reach efficient equilibria in a simple, *autonomic* fashion, without any external control
- ▶ The bottom line: draw inspiration from nature to build a distributed autonomic system based on local interactions

THE BIONETS VISION

- ▶ Overcome device heterogeneity and achieve scalability via an autonomic and *localized peer-to-peer communication paradigm*
- ▶ Services are autonomic, and *evolve* to adapt to the surrounding environment, like living organisms evolve by natural selection
- ▶ Network operations will be driven by the services, providing an ad hoc support when and where needed to fulfill users requests
- ▶ The network will become just an appendix of the services, which, in turn, become a mirror image of the social networks of users they serve

Next Generation Grids and Global Computing are also composed of large sets of heterogeneous and intermittently connected equipments

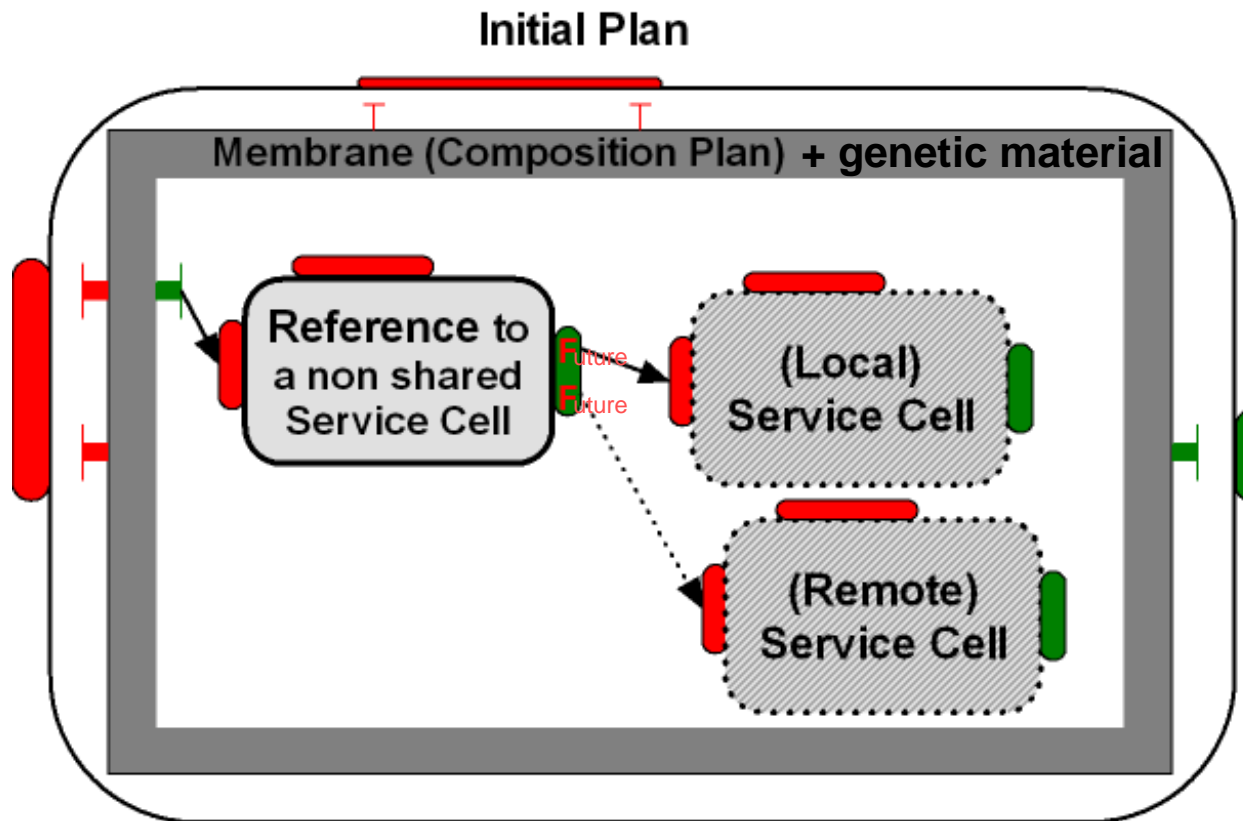
Services should drive the Grid ... not the contrary: disappearing Grid

An evolutionary service architecture as defined in BIONETS could fulfill NGG's requirements

Evolutionary Computing

- Classical: synchronous, centralized, offline search/optimization: 1 probl.
 - – Population of individuals: individual “genotype” = candidate solution
 - Evaluate candidate: Fitness
 - Selection of best candidates
 - Transformation: crossover, mutation: next generation
 - Optimum: Output: Stop
- BIONETS: asynchronous, decentralized, online self-optimization (autonomic)
 - Service population: alternative (competing) individuals
 - Cascade serv. provisioning & evaluation: graph
 - Resilient to transformations: on-line impact of mutations
 - Open-ended: Non-stop

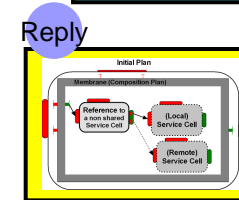
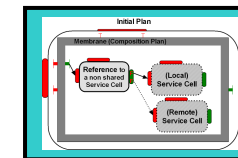
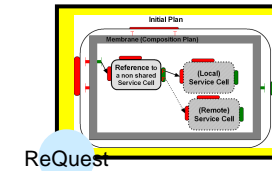
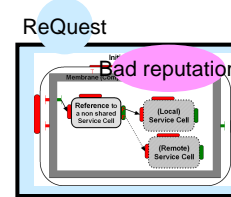
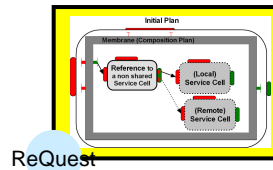
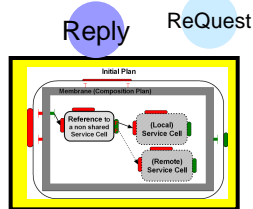
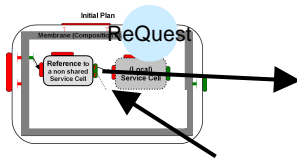
One possible support for a Bionets composed service



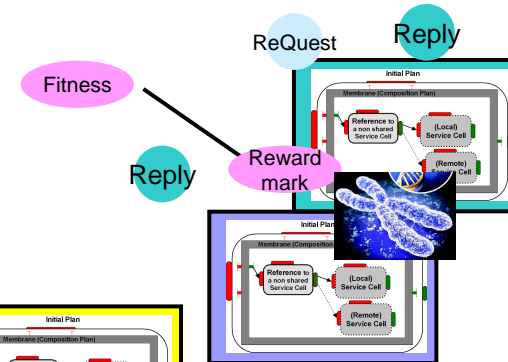
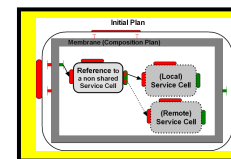
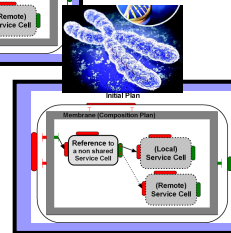
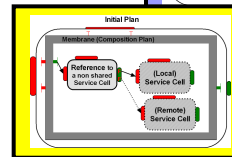
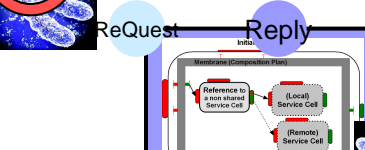
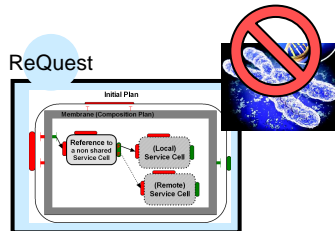
- 1) The initial plan is parsed and initiate the creation of a composite service as a composite hierarchical component
- 2) Inner service cells and bindings are adapted to the Bionets service particularities: discovery of services and virtual bindings, including the fact that replies may come very late or even never (resilience to this can be tackled by 'futures')

BIONETS Service Discovery, Execution, Evaluation, Evolution, Depreciation

Global behaviour:
comes from the
large
number of mobile
devices
(statistical behavior)

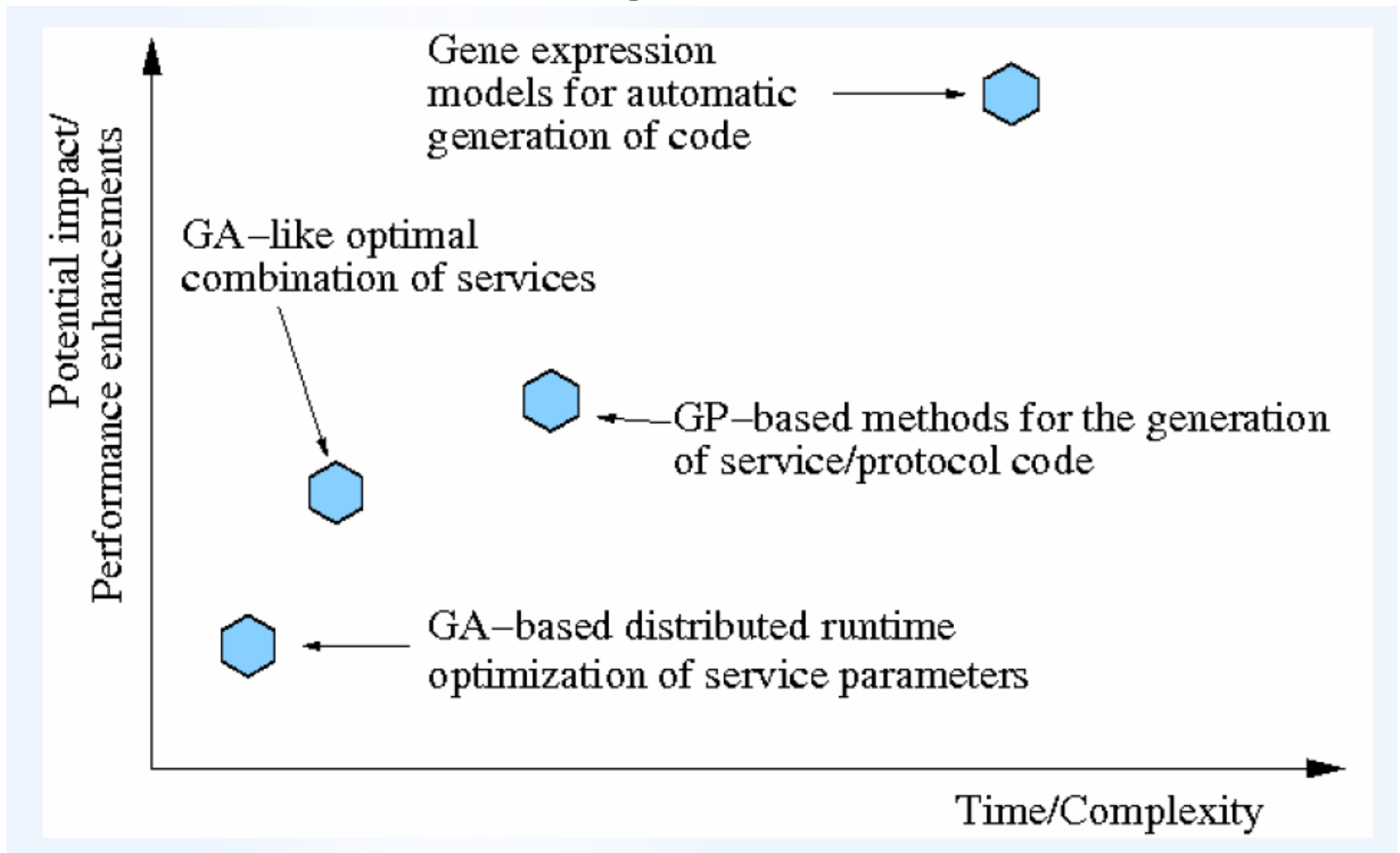


Becomes
deprecated,
so either
self-evolve
or dies



Bio-inspired (genetic, chemical, computing) evolution
But also socio-economic model,
evolutionary multi-players game theory, swarm intelligence

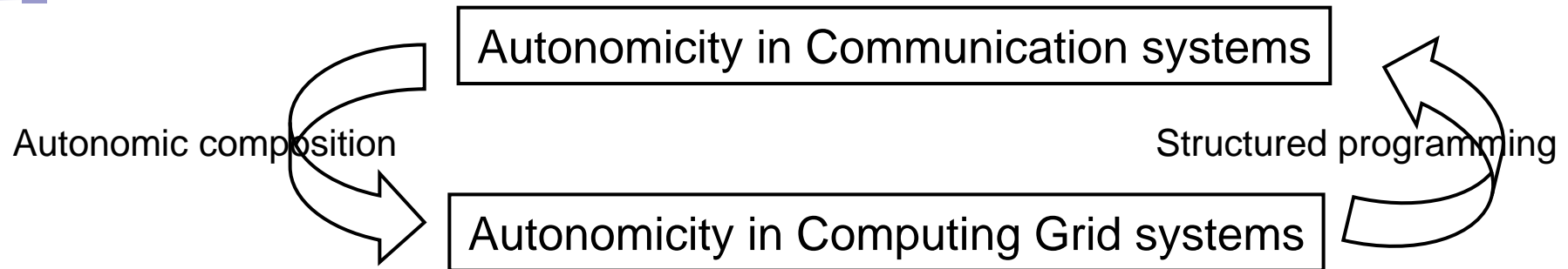
Bio-inspired (genetic) evolution





Wrapping up

- We focused on application level services
 - But component-oriented, structured approaches are also needed to support the environment, middleware, ...
 - Autonomic Application Servers (e.g. INRIA Jade tool for autonomic management of J2EE AS),
 - Autonomic Workflow Engines (e.g. ETHZ extended JOpera)
 - ... for which structured autonomic elements, like software components, can be useful also
- Autonomic features at architecture & programming levels:
 - a vertical concern, from high-level design down to middleware



- Having Autonomic Composition of Services in the context of autonomic communications systems is very demanding
 - Discovery space fully open, no support except locally on the device
 - extreme degree of self-organization, self-optimization, self-management
 - Solutions could then be reused in more closed worlds (like in classical grid systems)
- Software component models in distributed Grid computing
 - structured, hierarchical and distributed grid components as the CoreGrid GCM, *“How to program the GRID ?”*
 - Strong encapsulation, raising the level of abstraction
 - clear separation between functional and non-functional concerns
 - Strong support for autonomic behaviour support in SAC projects