



Spontaneous grids over the Internet (or how to set up a Digital Ecosystem)

Miguel Vidal
Sun Microsystems
miguel.vidal@sun.com
DBE Technical Coordinator

Fostering the Research and Technological Development, the DBE Project is an Integrated Project presented under the first call of the VI EU Framework Programme. The initial activities have started in November 2003.

What is the DBE Execution Environment?

- ◆ It is the nervous system that captures run-time behavioural data of the DBE users, connects them to the DBE infrastructural components, and mediates adaptive reactions in useful time
- ◆ It represents the short-term memory of the DBE in its dynamic network topology
- ◆ It is the supporting infrastructure for the DBE Evolutionary Environment
- ◆ It is an environment that allows the transposition of LAN behaviour to the Internet
- ◆ It is a dynamic, distributed, resilient, P2P platform that allows the sharing of services between peers
- ◆ It is a network architecture that implements a broadcast layer that enables real-time search and discovery and event handling over the Internet

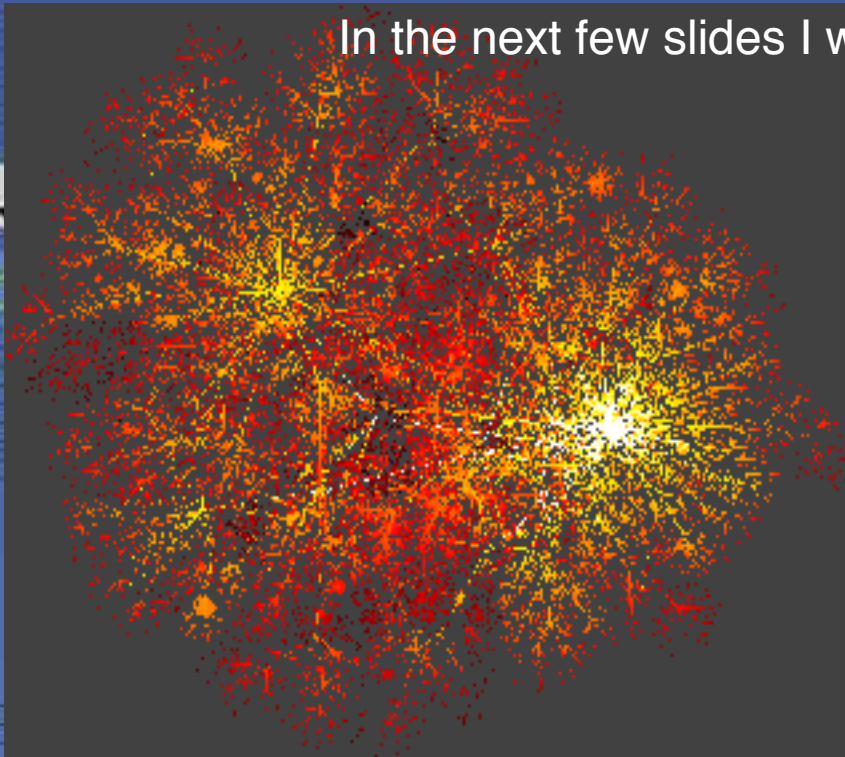


The Role of the Network

If we look at Web Services today, it is difficult to imagine how we could achieve the functionality summarised in the previous slide

Our approach has been to recognise the quantum leap in performance, reliability, QoS, and ease of use that could be achieved if a few key innovations could be engineered **at the same time** to define a new paradigm of network architecture

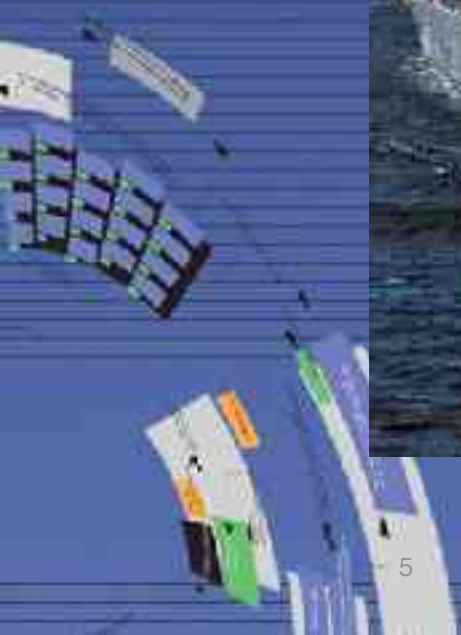
In the next few slides I will summarise what these innovations are



Why? (a technical answer)



The Jini Battleship



F1 Real time Telemetry System



Why the Jini Approach?

- ◆ Jini gives greater flexibility in the integration of components because it allows the movement of code, in the form of the proxy
- ◆ Web Services do not provide the code, they only provide the interface. Thus, the provider and the consumer must implement the same interface
- ◆ In the DBE, only the provider needs to implement the code. The consumer only needs to download the proxy.
- ◆ Jini enables the dynamic discovery of services



FADA: Federated Autonomous Directory Architecture

- ▶ Because Jini only works in a LAN we re-implemented it from scratch for the Internet (2002, FETISH FP5 Project)
- ▶ The requirement to estimate the proxy lease time accurately required us to solve the well-known latency forecast problem of the Internet. We did this using Kalman filters with 95% accuracy
- ▶ Because the Internet is a scale-free network, the signal propagation time scales less than linearly. Realising this motivated us to implement a dynamic search and discovery mechanism over the Internet
- ▶ This means that I do not need to know beforehand the URL of the service I want to consume, as is the case with Web Services
- ▶ This means that we are able to broadcast a query in a medium that does not support broadcasting
- ▶ This means that we could afford to go beyond Jini and define a distributed directory for the registration of services



Integration by means of '*standard*' Web Services

- ◆ Too many constraints (famous port 80 restriction).
 - ▶ Enforce SOAP data exchange (XML over HTTP).
 - ▶ A consensus needed to provide semantics to each interface
 - ▶ It is a RPC approach (the execution is delegated in the service provider)
 - ▶ The service can't be a piece of code
 - ▶ It is still tool dependent
 - ▶ Poor flexibility for security
- ◆ WSDL is not dynamic enough.
 - ▶ Dynamic IP
 - ▶ Multiple end-points
- ◆ UDDI was not designed for dynamic IP
- ◆ It is not self-managed: If you change something, just let me know
- ◆ It is not resilient by construction (self-healing).



I Have a Dream

- ◆ Imagine a world where computers fix their own problems before you even know something is wrong.
- ◆ Imagine you can use self-managing computing systems to control an increasingly complex and expensive IT environment.
- ◆ Imagine making your systems resilient, responsive, efficient, and secure without human intervention.

```

    .001.^
    u$DN=1
    z00BAI
    l..=^
    ;s<^
    NRX^=-^
    z0c^CX^
    ^B0s^^
    @0$H^
    n$0=XN;.^
    iBB0vU1=^^
    ^$000cRr^vul
    FAHZuqr-^
    ZZUFA0FI.^
    ;BRHv n$U^
    ^ARN1 ^@si
    ^Onv^ 01.^
    c0qr rs.^
    aUU^ ul.^
    ^RO- :.^
    nn^^ =.^
    =1^ .. ^

```



Digital Business Ecosystem main components

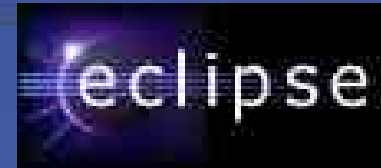
DBE portal

<http://www.digital-ecosystem.org>

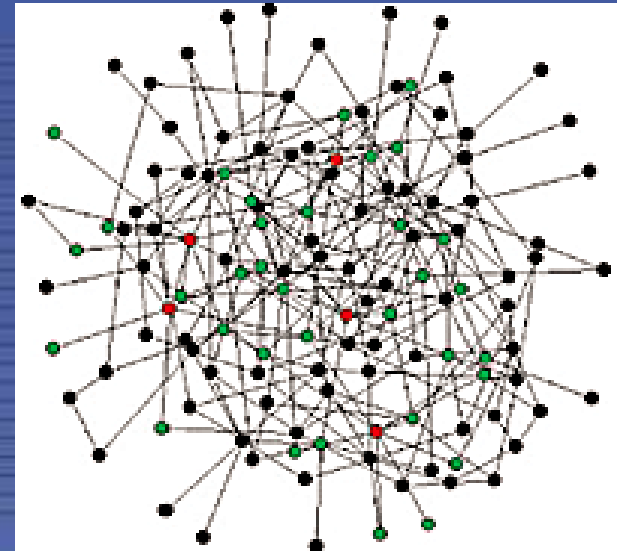
Plugins eclipse: BML,SDL,...



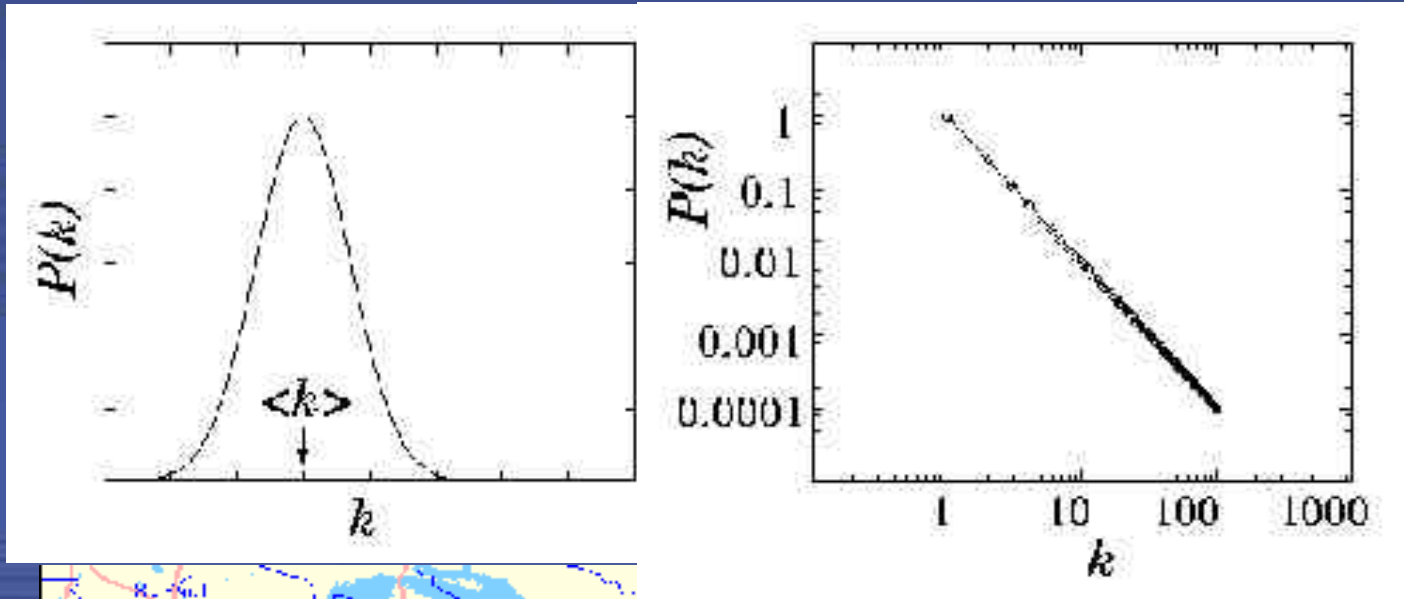
Service Factory



Execution Environment



What does Scale-free mean?



Exponential Network

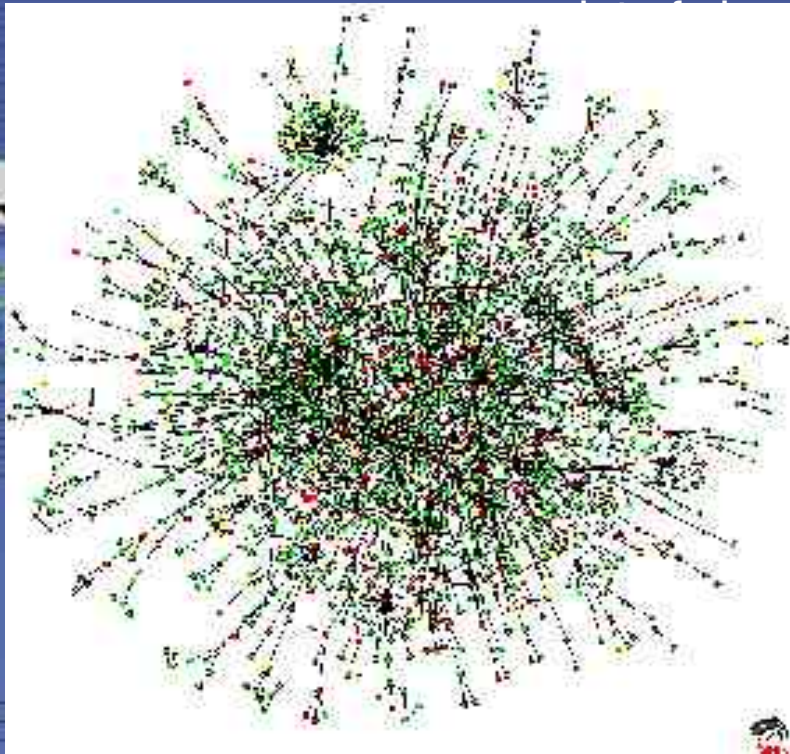


Scale-free Network

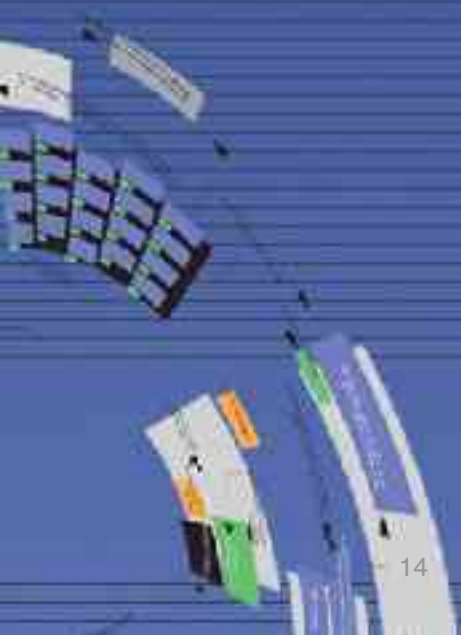
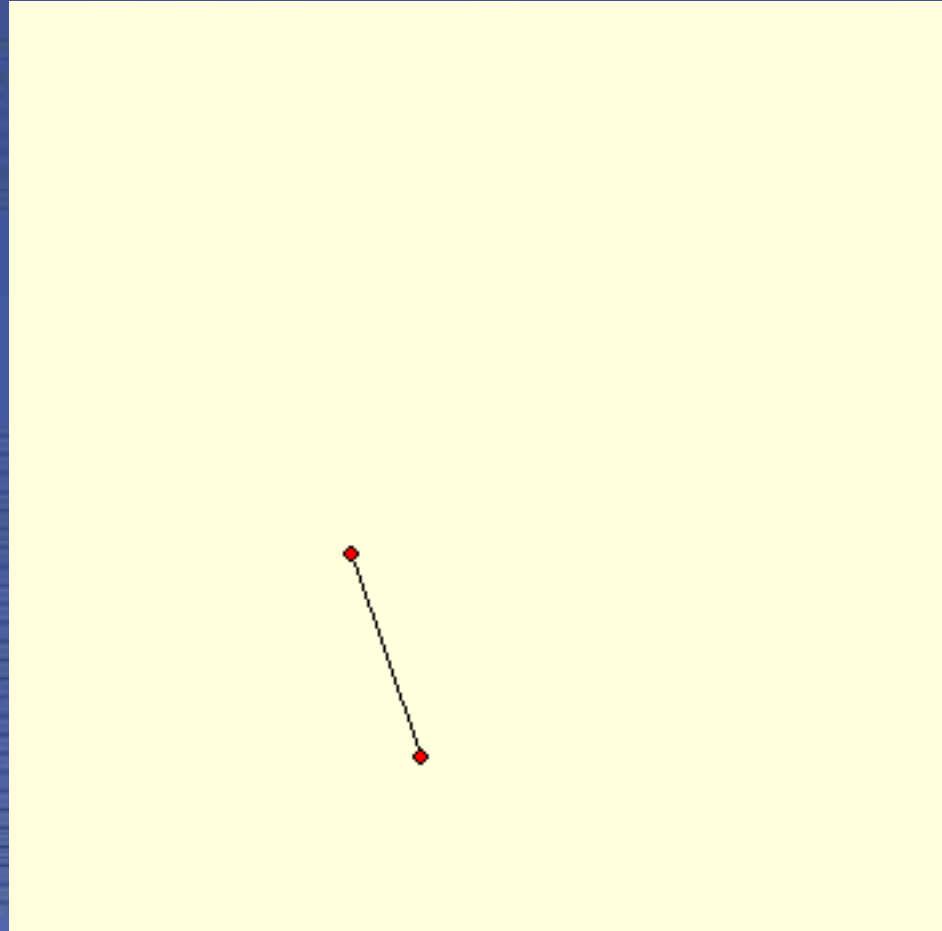


Scale-free Networks Are:

- ◆ Robust
- ◆ Degrade gracefully in the presence of node failures
- ◆ Their structural and dynamic properties can be quantified accurately and reliably through well-understood mathematical models
- ◆ More importantly, these properties remain invariant with the growth in size (scale) of the network
- ◆ This is the most important property from the engineering

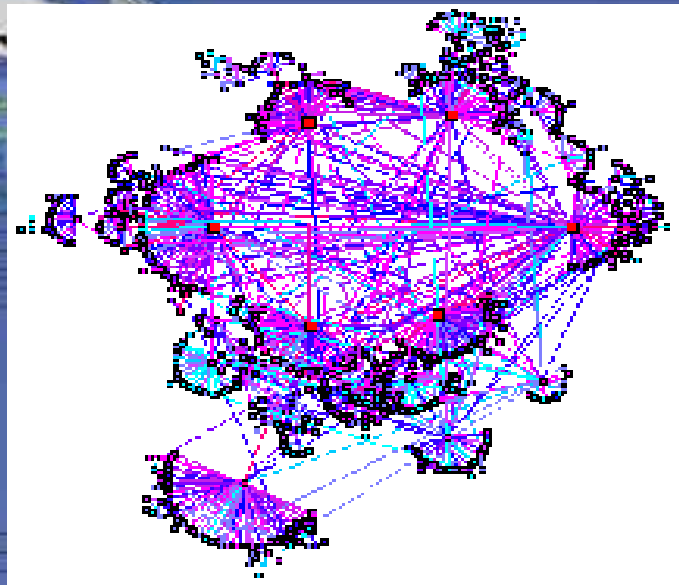


The sustainable growth model



The ServENT

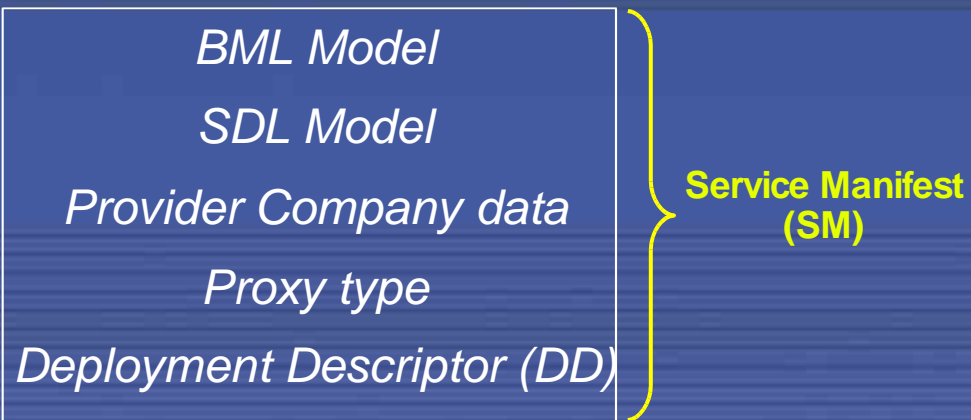
- ◆ The servent is in charge of all DBE communications and execution because it is the container where all the services are deployed
- ◆ The ServENT is the main actor in the modular DBE architecture, whose many levels of indirection allow us to manage complexity behind the scenes
- ◆ Developing our own “container” (as opposed to Tomcat) allows us to do what we want and what we need, without restrictions
- ◆ The network of ServENTs is the infrastructure of the Digital Ecosystem!



- P2P architecture
- data synchronization
- authentication
- self-healing
- service migration
- ...



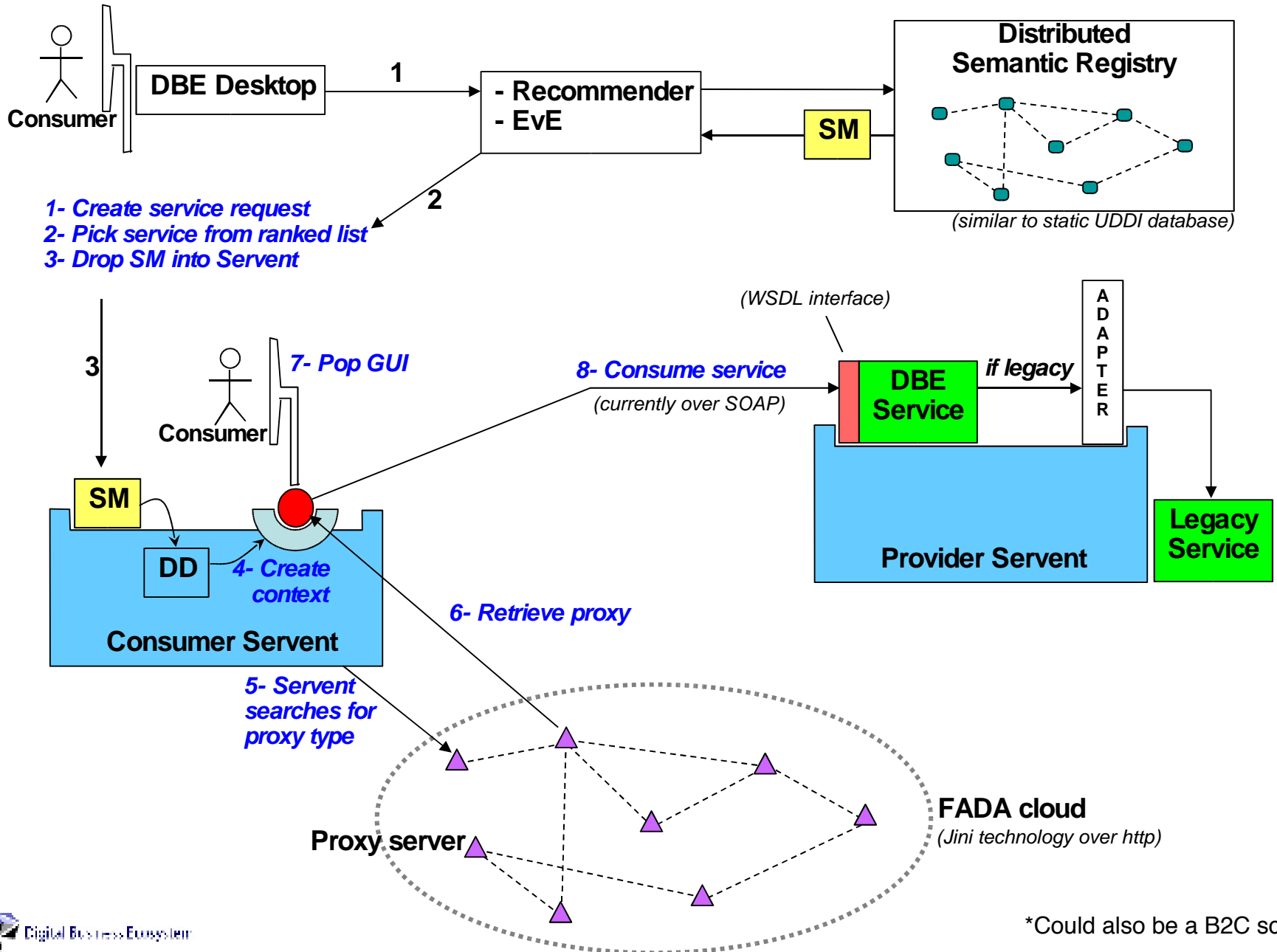
Service Manifest



Servent = SERver + cliENT

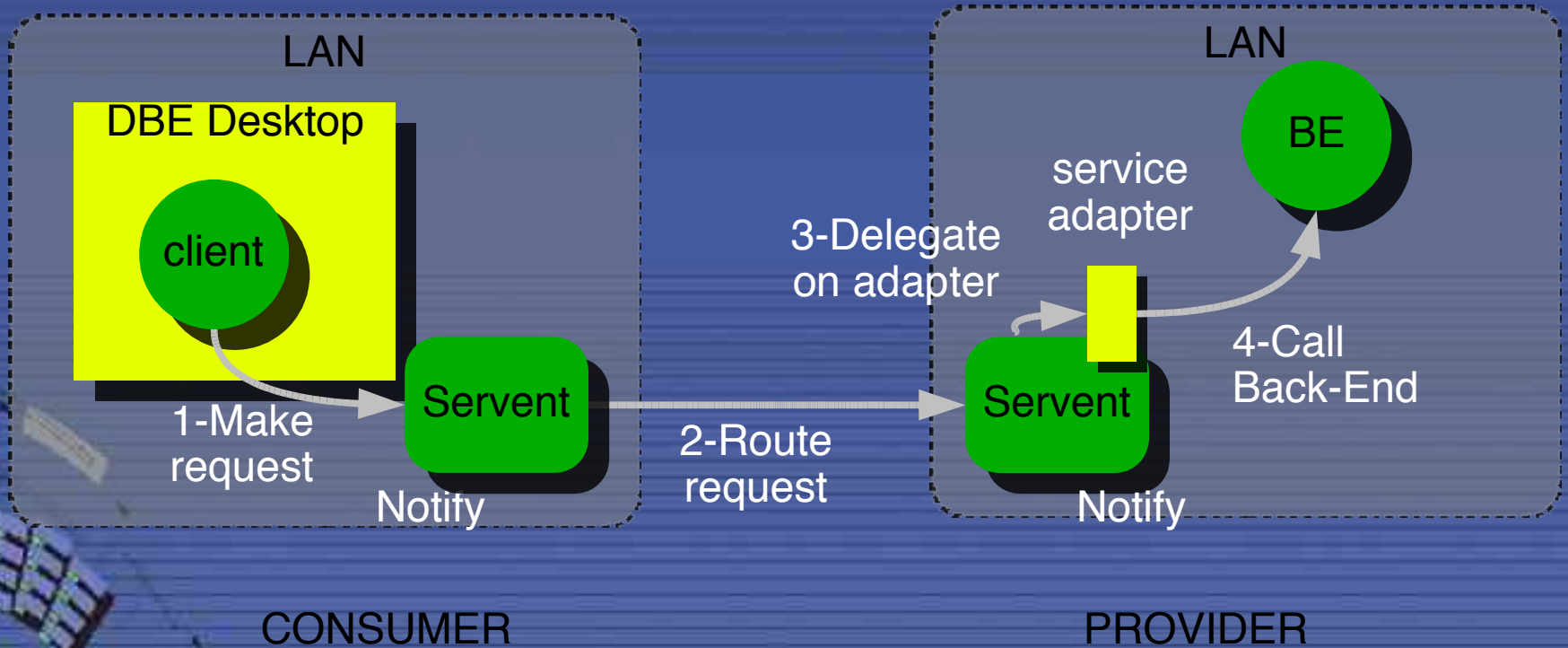


Consume Service*: Behind the Scenes

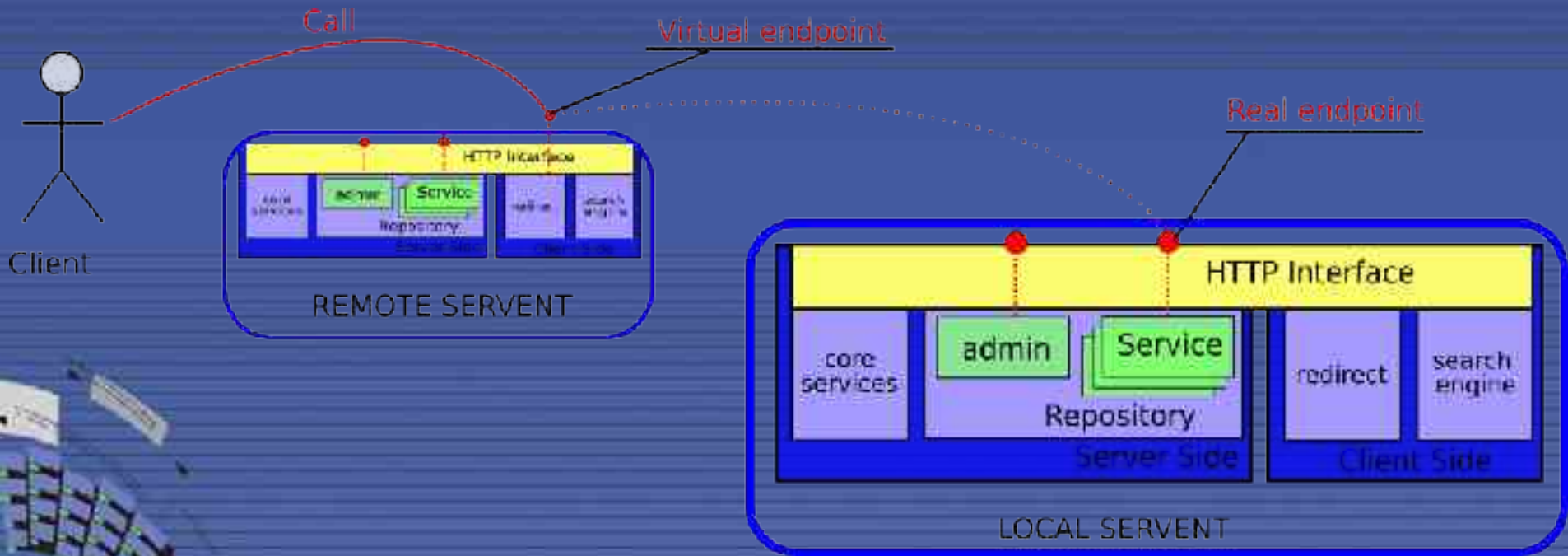


*Could also be a B2C scenario

DBE Server + cliENT (ServENT): Service invocation



DBE SERVENT in detail



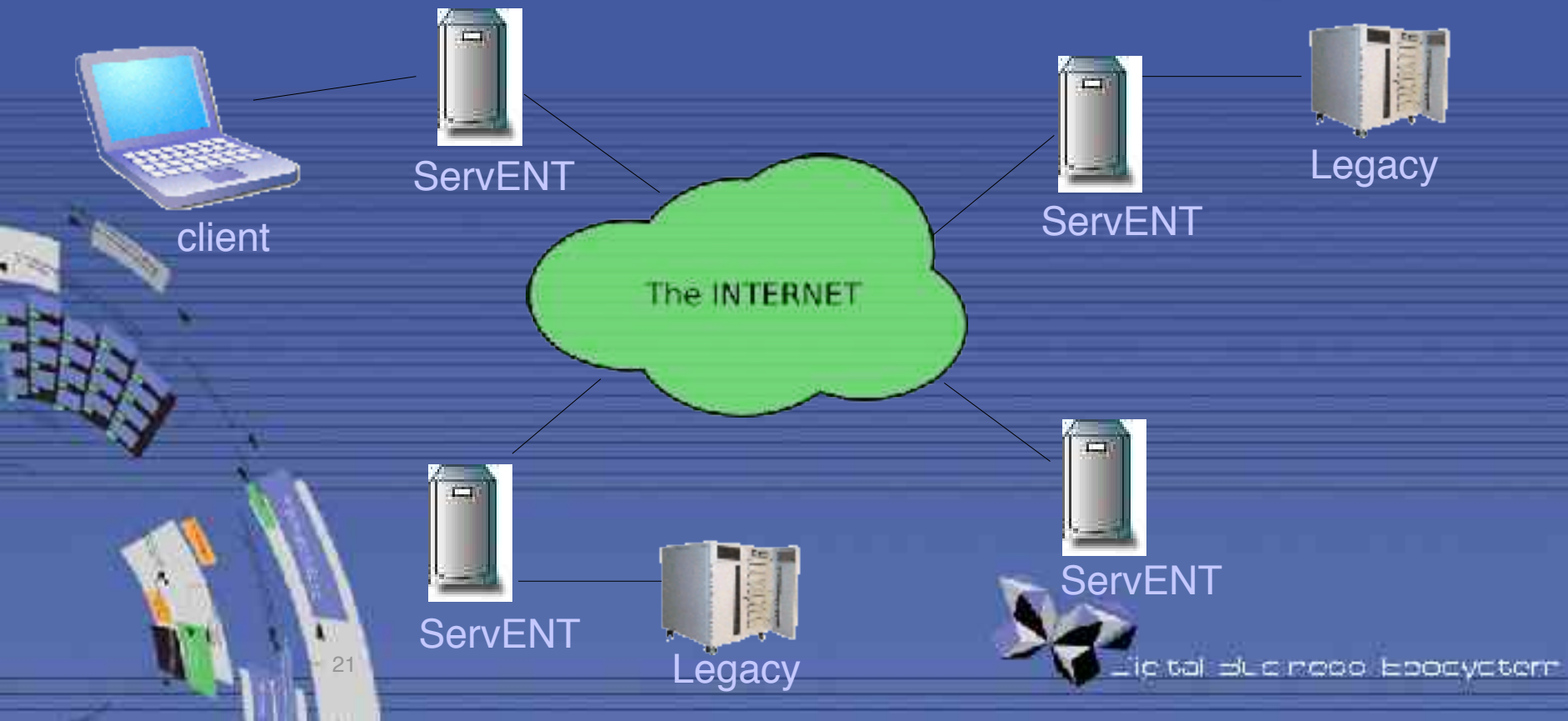
The Demo

- ◆ The demo concentrates on the Execution Environment
- ◆ The demo covers only steps 3-8 of the previous slide
- ◆ Because we cannot yet demonstrate the entire service creation and consumption life-cycle, we decided to show how the DBE infrastructure can be used to connect between many Personal Area Networks (Bluetooth) across a Wide Area Network
- ◆ An important point which is not necessarily evident from the demo is that the connectivity is based on dynamic Ips, and therefore fully supports user mobility
- ◆ The demo consists in sending a multi-media message from any peer to any Bluetooth device that supports the OBEX profile
- ◆ We believe that this is a credible demonstration of how any user who only has access to a mobile phone can take advantage of the power of the DBE



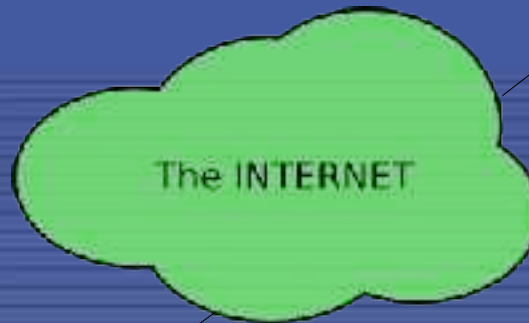
DBE ExE Architecture

- ◆ Many interconnected ServENTs around the world offer services to clients
- ◆ Some services are new and others are old legacy systems



Easy distributed development

- ◆ Services are deployed locally but they can be used remotely



- ◆ This remote communication is done automatically without extra programming



Highlights: From a PAN to a WAN using DBE ExE

- ◆ Legacy applications based on Bluetooth deployed on DBE
- ◆ DBE infrastructure allows SMEs the use and development of new distributed services
- ◆ P2P architecture without single point of failure or control
- ◆ Technology adapted to the user's habits: mobile phone, usability, pervasiveness, ...
- ◆ High scalability and robustness (more peers means more resources)
- ◆ The information arises and is consumed at the edge of the network

