



Australian Government

Department of Defence

Defence Science and
Technology Organisation

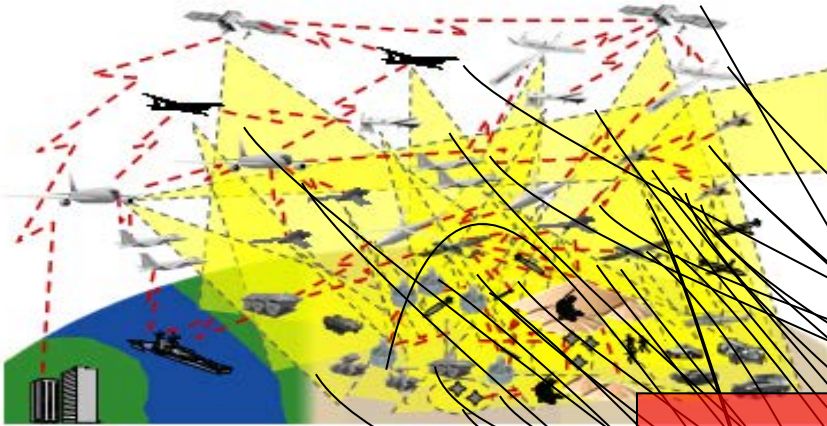
A Common Service-oriented Infrastructure Approach for Defence Tactical Environments

Derek Dominish

Aerospace Division
Future Information Architectures

OMG – Berlin 17th June 2013

The NCW Integration Complexity Problem



Problem Space

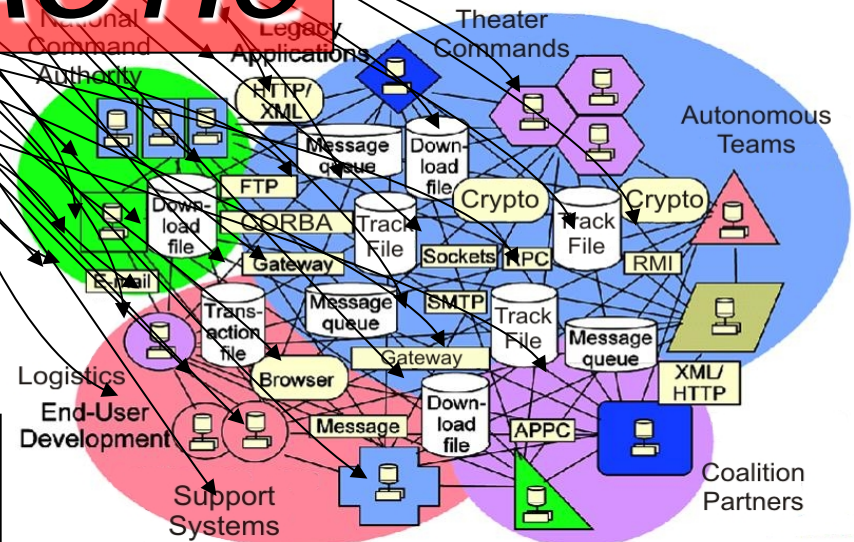
- Network-centric, dynamic, very large-scale “systems of systems”
- Stringent simultaneous quality of service (QoS) demands
- Highly diverse & complex problem domains

CHAOTIC

Solution Space

- Enormous accidental & inherent complexities
- Continuous evolution & change
- Highly heterogeneous environments

Mapping & integrating *problem artifacts* to *solution artifacts* is extremely difficult



DSTO

Open Architecture through Open Standards

Surface System



'POSIX', CORBA, DDS, etc.

Standards Based Computing Environment

TM, NAV, ID, etc.

Common Functions

Surface Combatant Unique Functions

SCS System Unique & Common Applications & Interfaces

Aircraft System



'POSIX', CORBA, DDS, etc.

Standards Based Computing Environment

TM, NAV, ID, etc.

Common Functions

Aircraft Unique Functions

Aircraft System Unique & Common Applications & Interfaces

Land System



'POSIX', CORBA, DDS, etc.

Standards Based Computing Environment

TM, NAV, ID, etc.

Common Functions

Land Unique Functions

Soldier System Unique & Common Applications & Interfaces

Sub-Surface System



'POSIX', CORBA, DDS, etc.

Standards Based Computing Environment

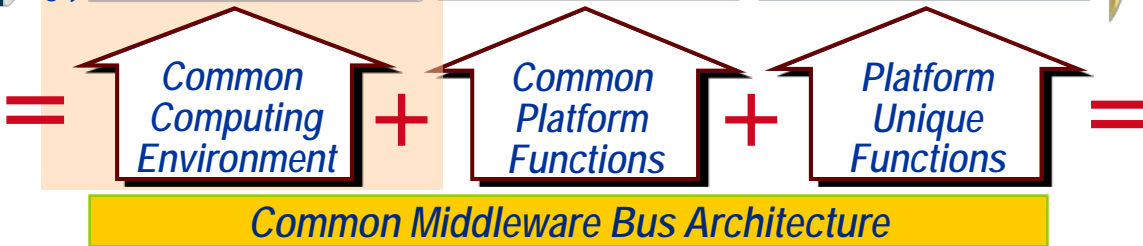
TM, NAV, ID, etc.

Common Functions

Sub-Surface Unique Functions

Submarine System Unique & Common Applications & Interfaces

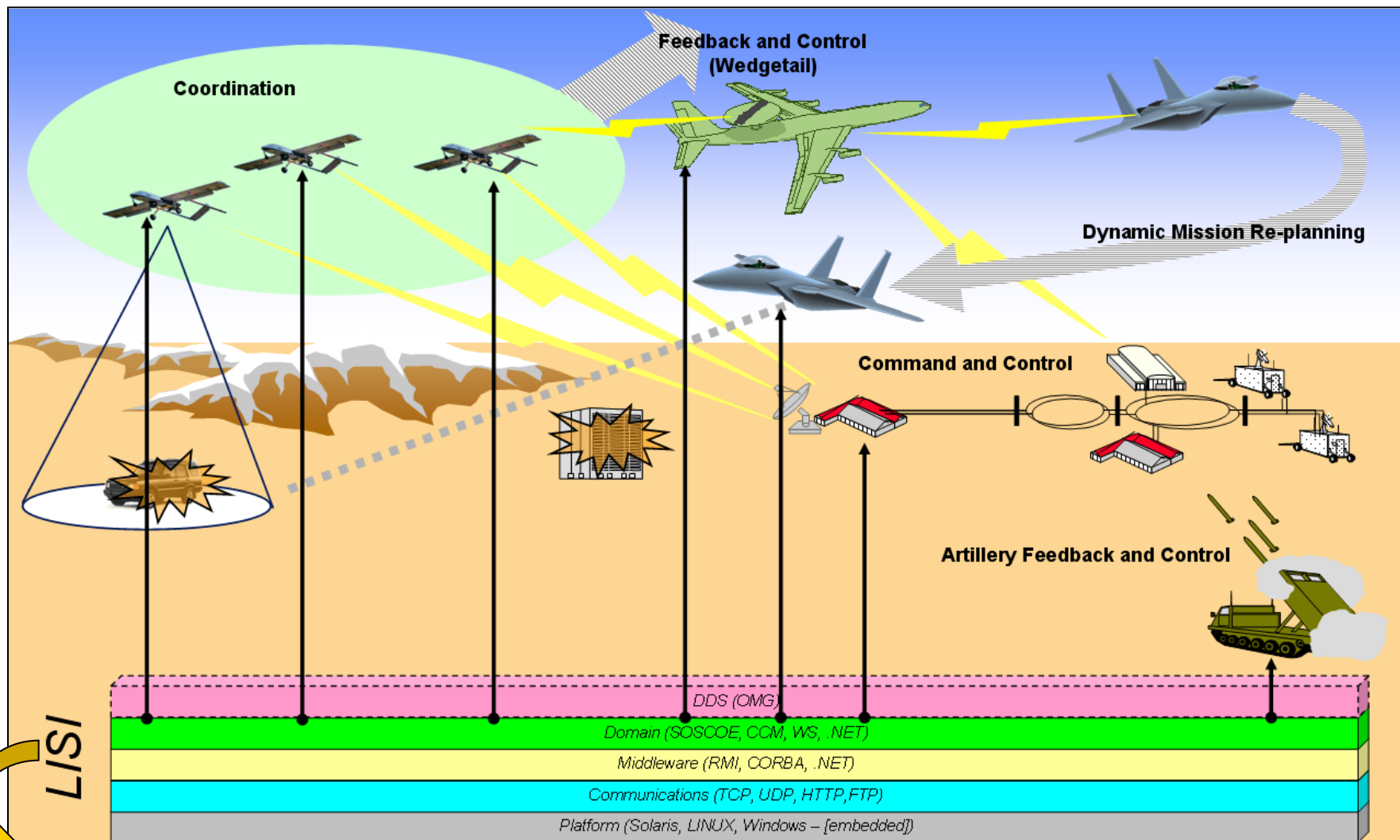
Service Orientated - Open Architecture



Interoperable System-of-Systems



Net Centric Mission Environment

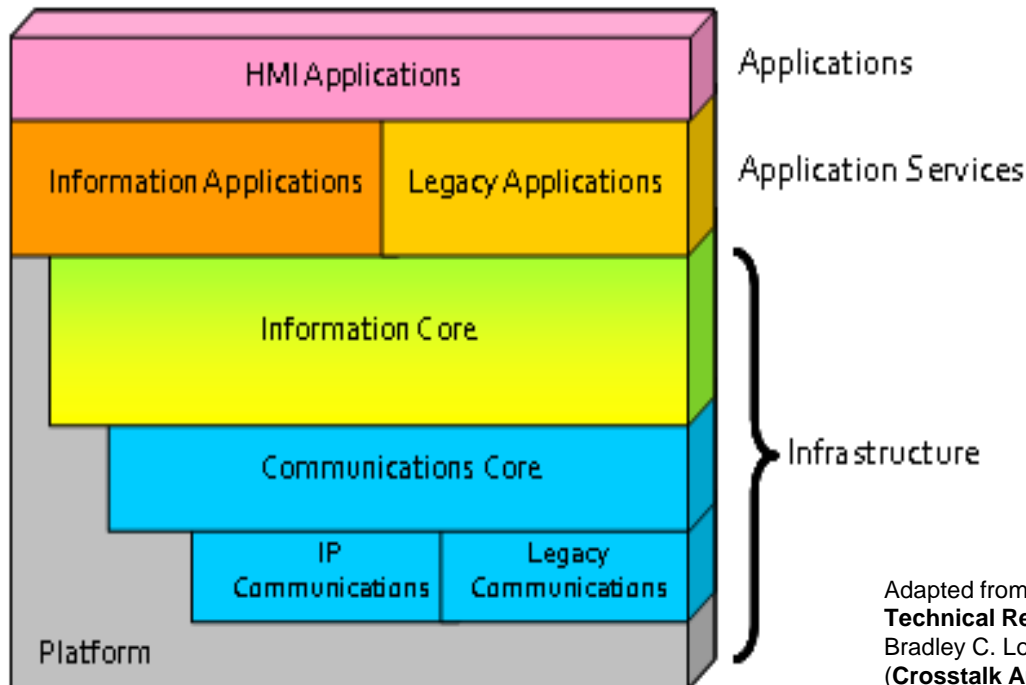


Levels Of Information Systems Interoperability (LISI) Reference Model



Layered Reference Architecture Tactical SOA (OA)

The Strategic Architecture Reference Model (SARM)



The Strategic Architecture Reference Model (SARM)

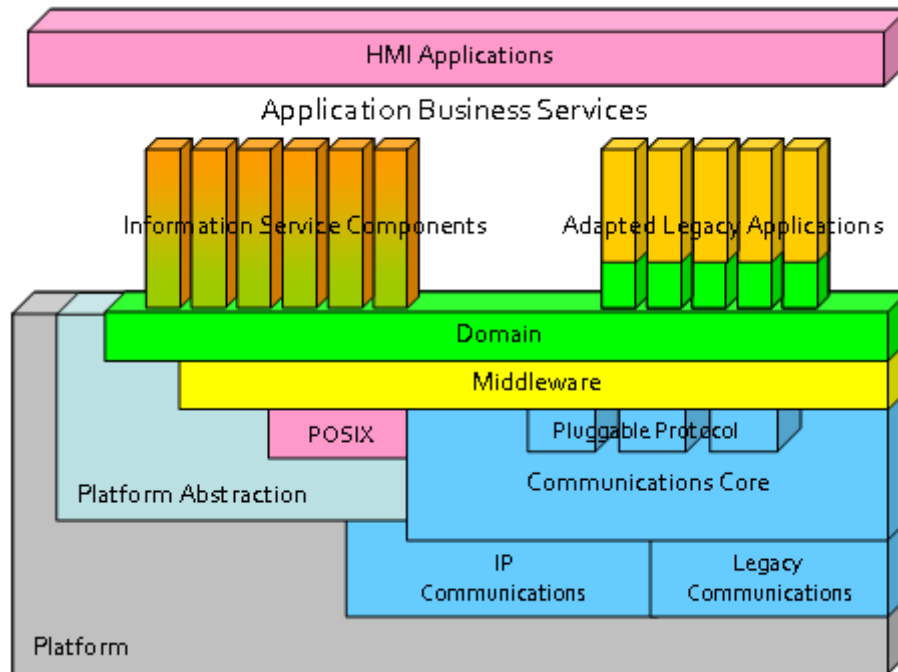
The SARM is;

- an open communications and information architecture framework supported by NCOIC,
- based upon commercial and government interfacing standards (IDL),
- organized to address system-wide network design issues,
- an enabling technology framework to allow platforms and tactical systems to interface to the Global Information Grid,
- allows for interoperable nodes on the network.

Adapted from:
Technical Reference Model for Network-Centric Operations
Bradley C. Logan. *The Boeing Company*
(Crosstalk Aug 2003; Vol16, No8)

Tactical SOA Reference Architecture

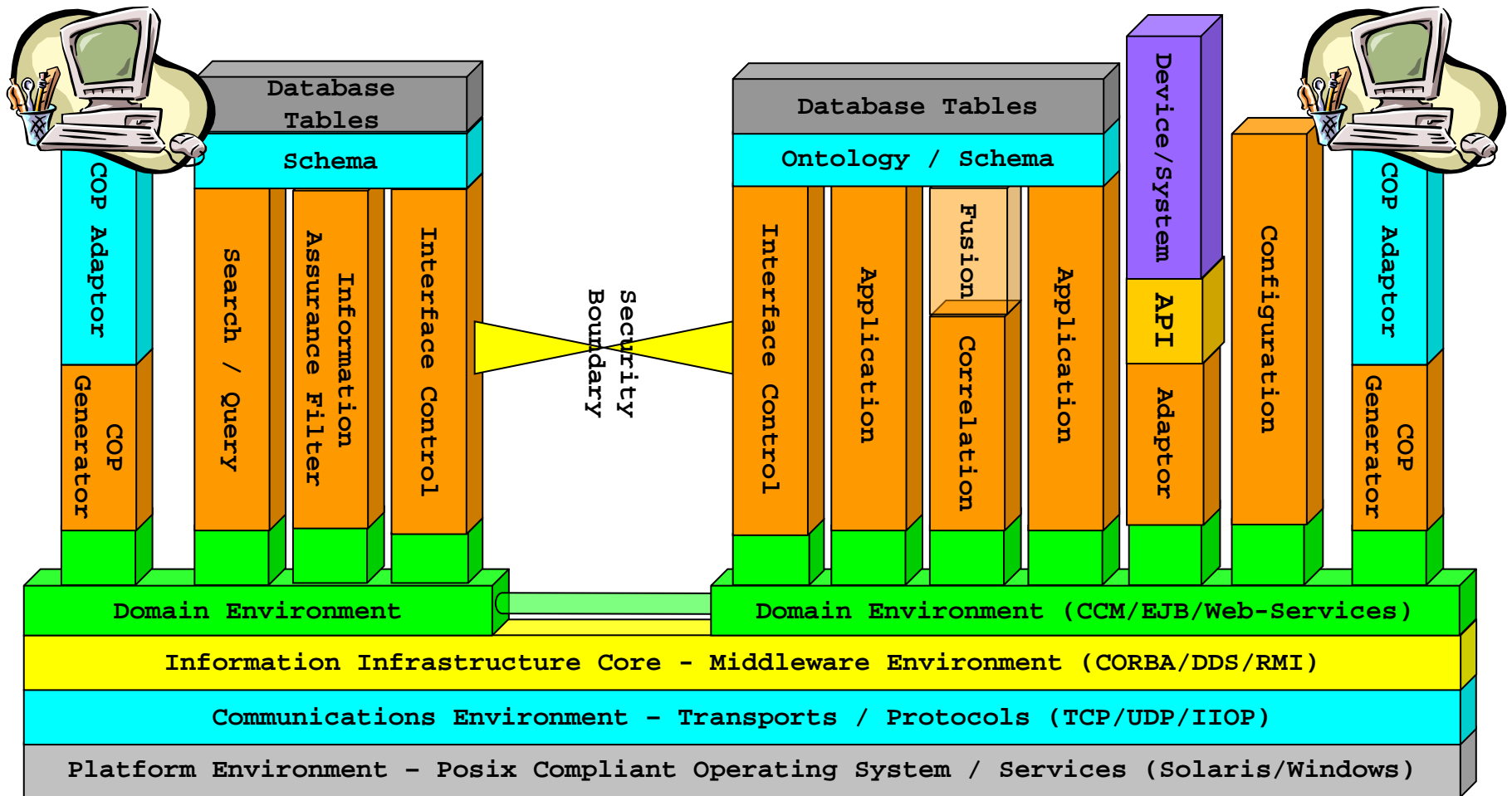
Business Services



Business Services are;

- a unit of autonomous behaviour that meets a particular business need,
- constructed / assembled from one or more components that support a business ontology,
- conform to an infrastructure plug and play deployment policy,
- orchestrated through an application,
- collaborate with other peer services within a network,
- location and machine architecture independent,
- built upon standardised infrastructure services and mechanisms,
- often needed to adapt to legacy environments,
- expressed through formal and interoperable service interface definitions (IDL [PIM]) to the network.
- not necessarily reliant on any particular networking protocol or communications infrastructure.

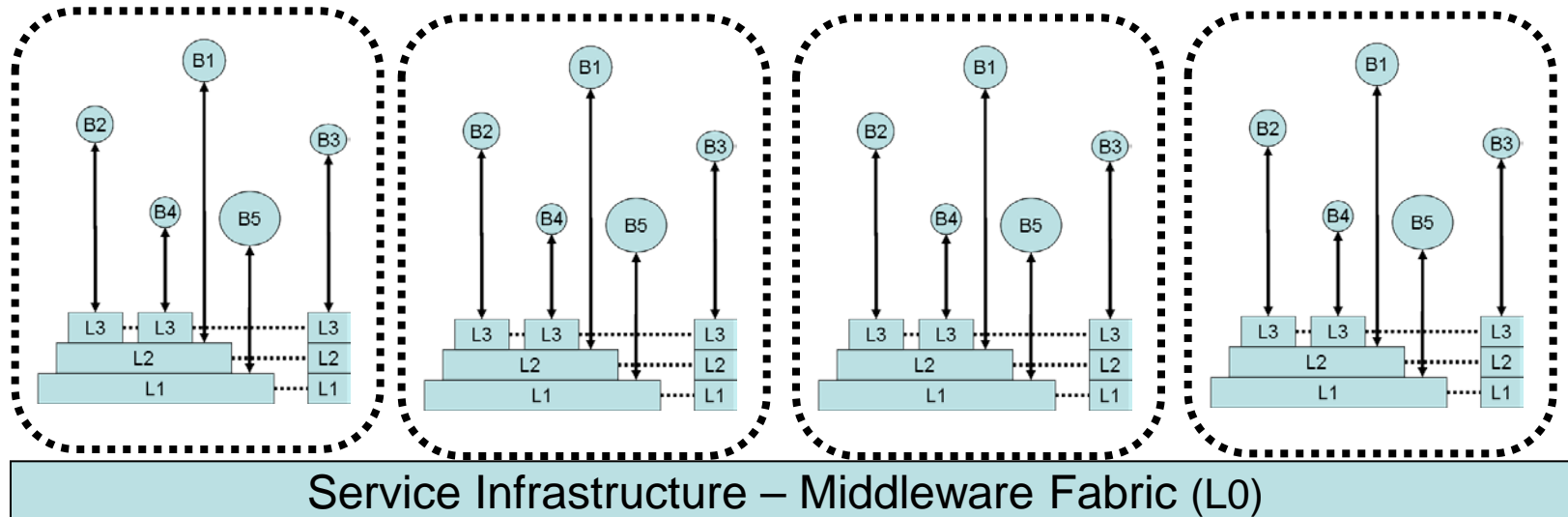
Net-Centric Reference Architecture (SOA)



System-of-Systems Integration Infrastructure Architecture

A Standardised Middleware Approach

System Integration Through Service Composition

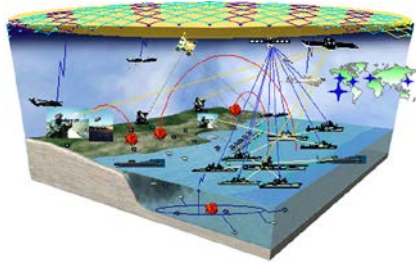


Information Assurance

- Communication: QoS, encryption, bandwidth
- Security: boundaries, classification, communication routes
- Authentication: information ownership, need to know

System-of-Systems Integration Infrastructure Architecture

A Standardised Middleware Approach



Interoperating across vendors and systems

- Ontological Conformity

Integrate “services” through layering

Standardisation

- Interfaces
- Protocols

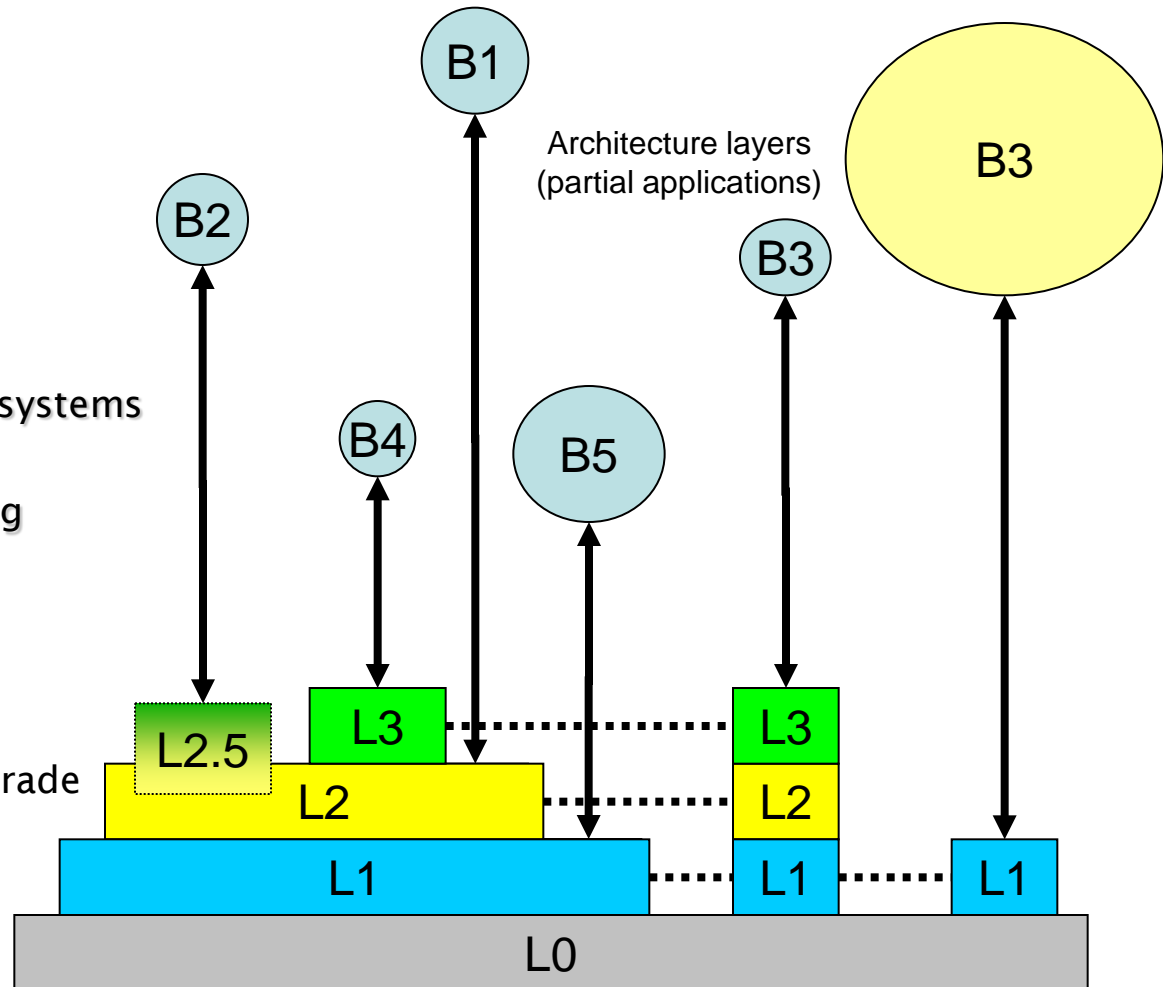
Hooking into appropriate layer

Easier, cheaper to change and upgrade

Open business practices

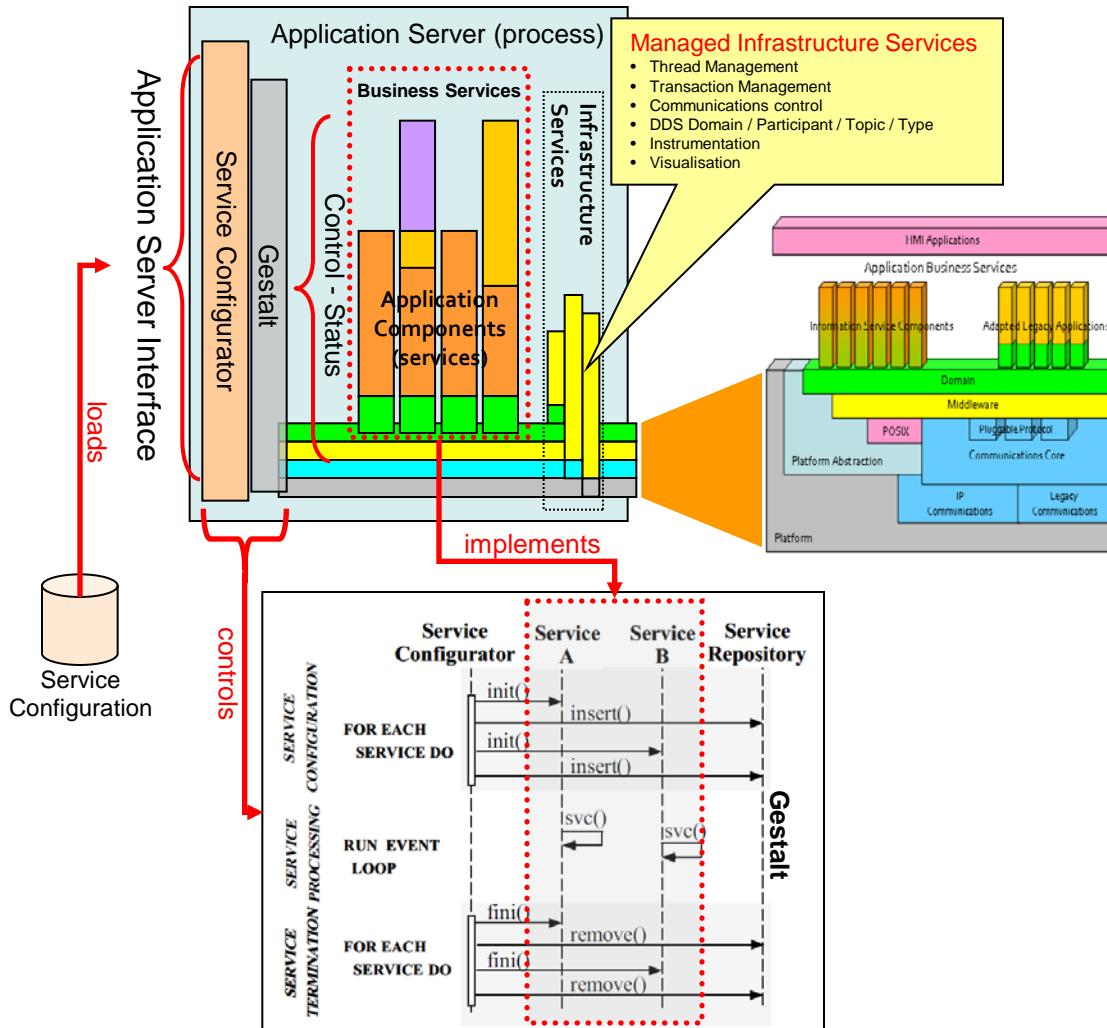
Indeterminate liability

- Certification



Application Server

'Application Store'

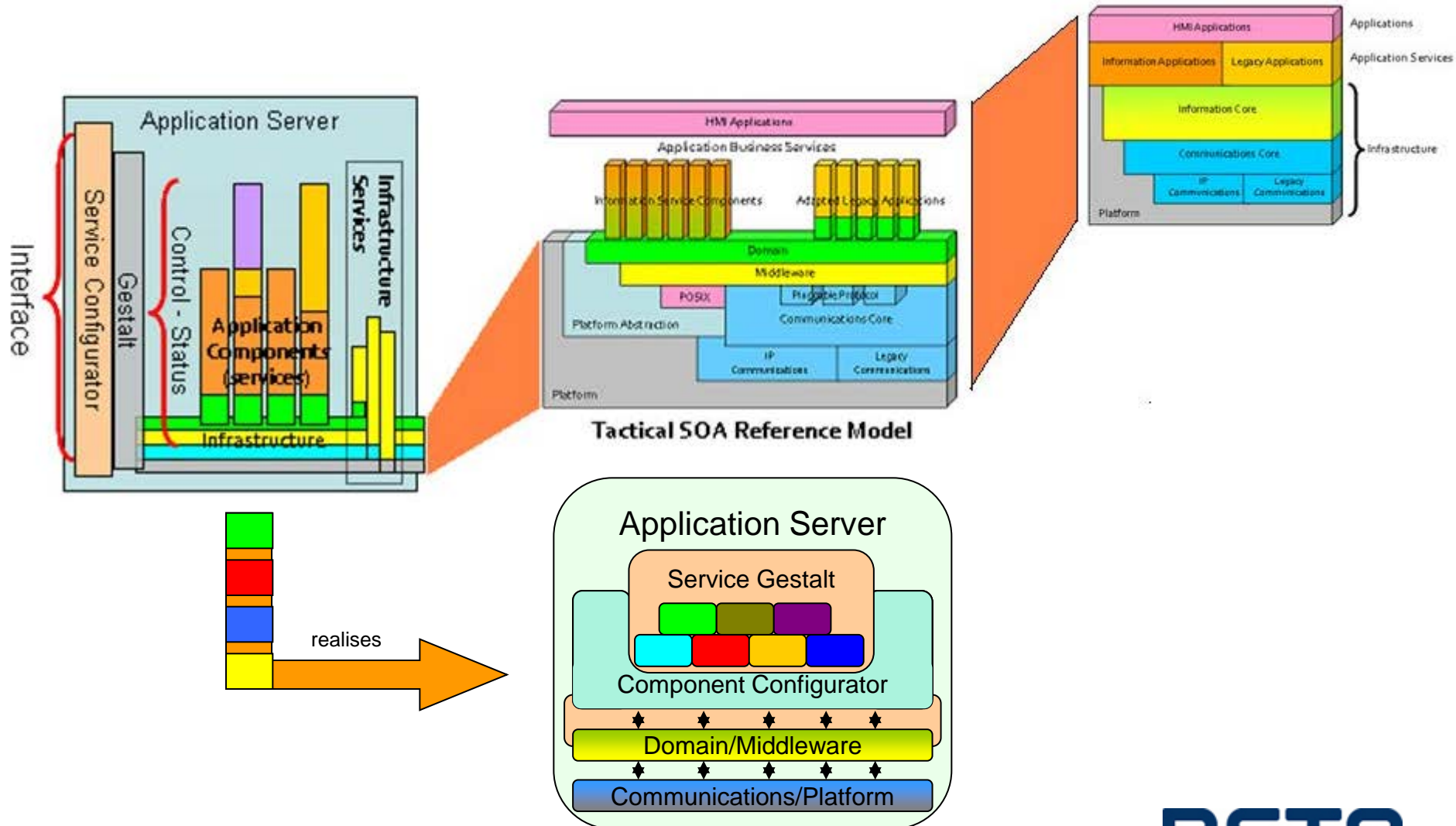


Application Servers;

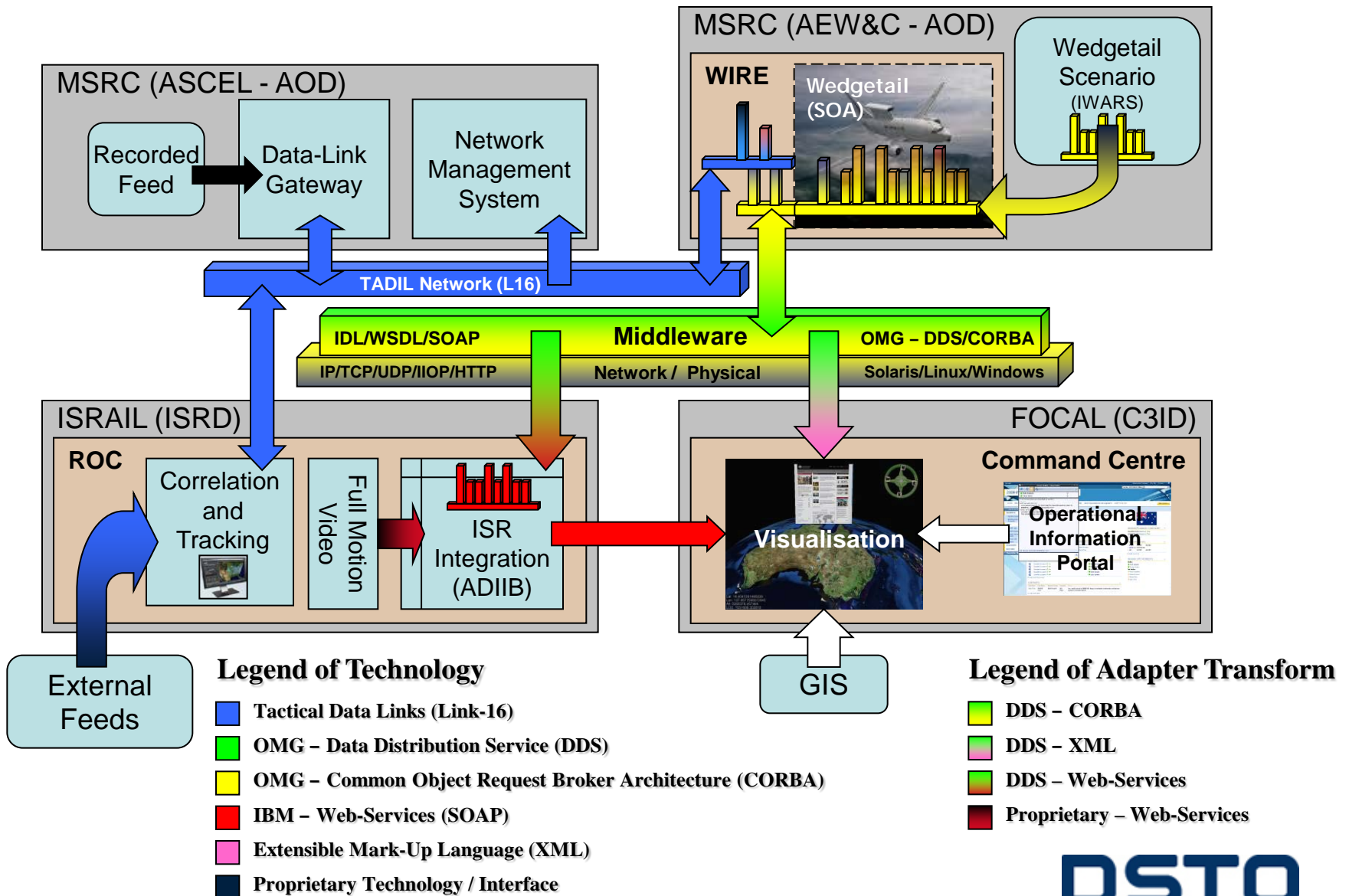
- manage component service complexity,
- dynamically and statically load and configure service components,
- provide infrastructure capabilities and services to hosted components;
 - logging,
 - aids to debugging,
 - instrumentation,
 - capabilities dynamically configurable and extensible.
- manage coupled middleware environment;
 - security and access policy enforcement,
 - Communications, protocols and bearers,
 - ... many other environmental aspects.
- manage lifecycle of hosted components (initialisation, shutdown, suspend, resume),
- facilitate the navigability to individual component interface implementations,
- manage the optimisation of peer component interactions through collocation,
- location and machine architecture independent,
- built upon standardised infrastructure services mechanisms and patterns,
- supports a network accessible service control and status interface.

Technical Architecture

'putting it all together'

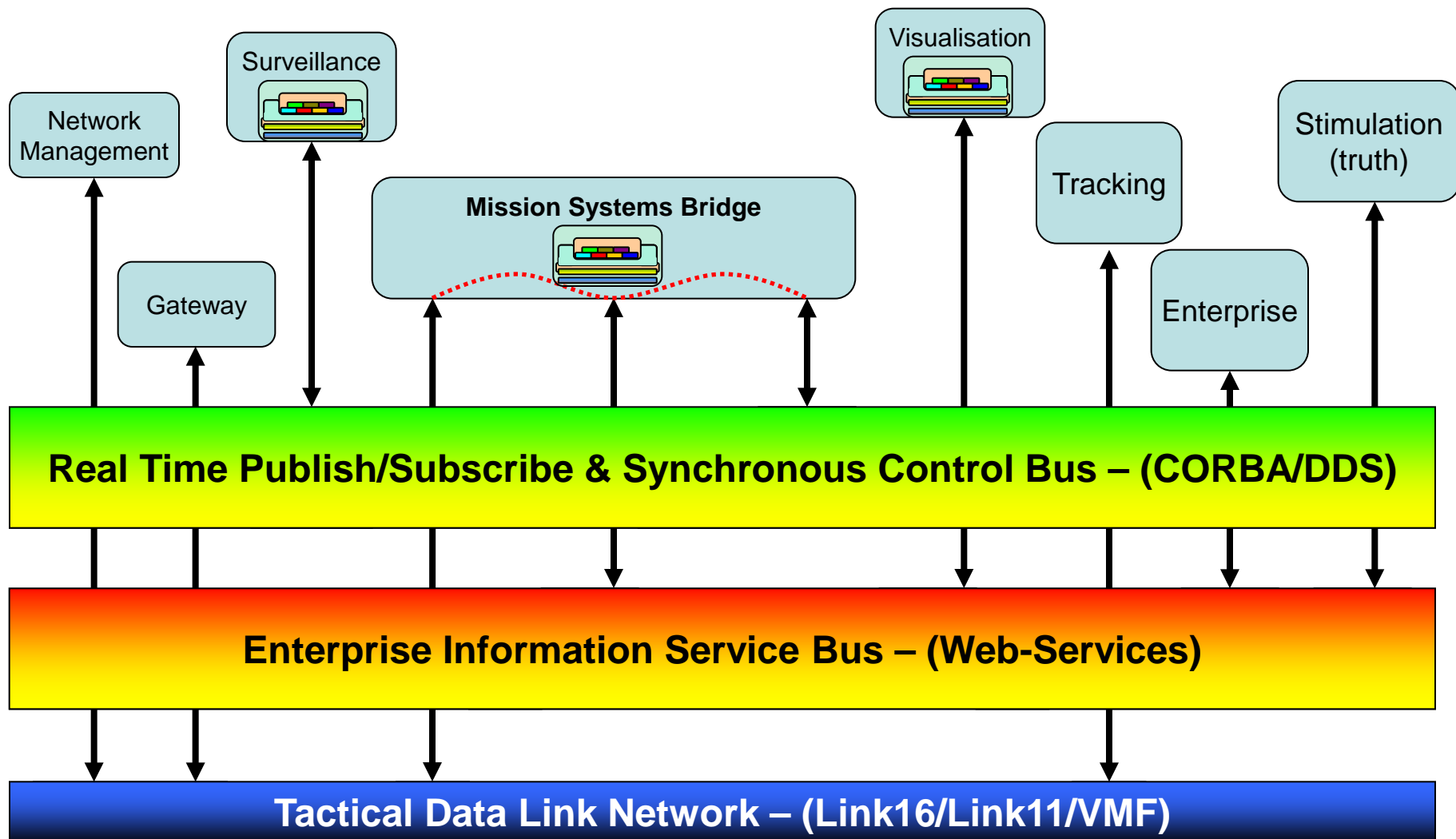


Net Warrior D10 Process Domain



Net Warrior (D10 Event) Bus Architecture

A Standardised Middleware Approach - Demonstration Event



Demonstration Artefacts – This really works!

The Build Process.

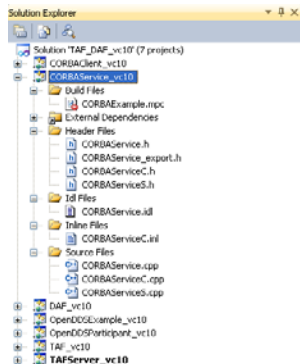
- Create a service project MakeProjectCreator descriptor for MPC toolset

```
project(CORBAservice) : tafilib, taolib_with_idl
{
  sharedname = *
  idlflags += -Wb_export_macro=CORBAservice_Export \
             -Wb_export_include=CORBAservice_export.h
  libout = $(DAF_ROOT)/lib
  libpaths += $(DAF_ROOT)/lib
  macros += CORBAservice_BUILD_DLL
  prebuild = perl $(ACE_ROOT)/bin/generate_export_file.pl CORBAservice > CORBAservice_export.h
  IDL_Files {
    CORBAservice.idl
  }
  Header_Files {
    CORBAservice.h
    CORBAservice_export.h
  }
  Inline_Files {
  }
  Source_Files {
    CORBAservice.cpp
  }
}
```

- Use MakeProjectCreator (MPC) to build solution files for platform toolsets (Windows/Linux ...).

```
perl %ACE_ROOT%/bin/mwc.pl -type vc10 -name_modifier *_vc10 -apply_project TAF_DAF.mwc
```

- Load project into toolset environment (i.e. Windows - VC10)



- Build binaries with toolset applicable to platform environment.

The Development Process.

- Define an interface (IDL)

```
module DSTO
{
  struct Structure
  {
    short    i;
    long    j;
    string   k;
  };

  exception test_exception
  {
    short error_code;
    string error_message;
    string status_message;
  };

  interface SimpleCallback
  {
    void callback_op(in string name);
  };

  interface SimpleServer
  {
    attribute long test_val;

    void register_callback(in SimpleCallback callback_ptr);

    long test_method (in long x,
                     in Structure the_in_structure,
                     out Structure the_out_structure,
                     inout string name);

    void test_callback(in SimpleCallback callback_ptr, in string txt);

    oneway void test_oneway (in string name);

    void raise_user_exception () raises (test_exception);

    void raise_system_exception ();
  };

  const string MY_SERVICE_OID = "MySimpleServer";
  const string MY_CALLBACK_OID = "MySimpleCallback";
};
```

- Implement interface providing service descriptors for application server deployment control (CORBAservice.cpp).

```
DAF_EXPORT_FACTORY_DEFINE(DSTO, CORBAservice);
DAF_EXPORT_SVC_DEFINE(DSTO, CORBAservice
, DSTO::CORBAservice::svc_name()
, ACE_SVC_OBJ_T
, &DAF_MAKE_SVC_FACTORY_NAME(DSTO, CORBAservice)
, (ACE_Service_Type::DELETE_THIS | ACE_Service_Type::DELETE_OBJ)
, false // Service not initially active
);
```

- Deploy with configuration (DSTO.conf)

```
dynamic DSTO_DDSExample Service_Object * OpenDDSExample: make_DSTO_DDSExample()
dynamic DSTO_CORBAExample Service_Object * CORBAservice: make_DSTO_CORBAservice()
```



Questions?



The Age – Australian Newspaper