

AIDA Review

August 27, 2004



Aida Review



Contents

- Context - Ron Chestnut
- Objective
- Requirements
- Overall Schematic
- Status
- Software Architecture, Demo
- Design Highlights
- Management and Administration
- From Here to Production
- Future
- Performance Analysis - Bob Sass
- Error Handling spin-offs - Ron MacKenzie



Objective

Enable rapid development of High Level physics applications and ad-hoc analysis for accelerator optimization, esp on Unix.

- Multi-platform
- Distributed
- Structured data
- Parameterized
- Multi-language (c++, java, Matlab)
- V. little programmer resource
- Allow physicists to write apps and analysis.
- Future
 - Existing tools underutilized
 - XML, Xquery



Objective

Solution Method

Easy to use API for distributed, structured, parameterized data exchange

... of the form commonly needed for high level applications

... which hides network and language binding, but helps new data sources be brought on-line very easily.



Requirements

- Characteristics of High Level Applications
- SLAC specific environment
- Where Aida fits in



Requirements

Characteristics of High Level Apps



Fairly infrequent data access



Fairly few discrete definitions of data required

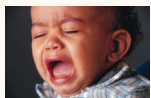
- definitions used repeatedly (so compile them)



Client generally does know the name and the type

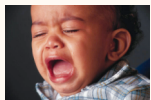


“User time” is ok. “Latency Appropriate”



Highly structured (“aggregates”)

- ‘array’ like, structured, arrays of structures



Parameterized



Requirements

SLAC Specific Environment

- Physicists want outrageously easy data access
 - Matlab (AT)
 - Model (>1 source: “SLC”, “pepoptics”), History Data
 - C++ (lego), Java/XML (XAL)



August 27, 2004



Aida Review



Requirements

SLAC Specific Environment

- Programmers stretched to the max
 - ∴ Minimize exposure to CORBA
 - ∴ Make adding Data Source really easy



August 27, 2004



Aida Review

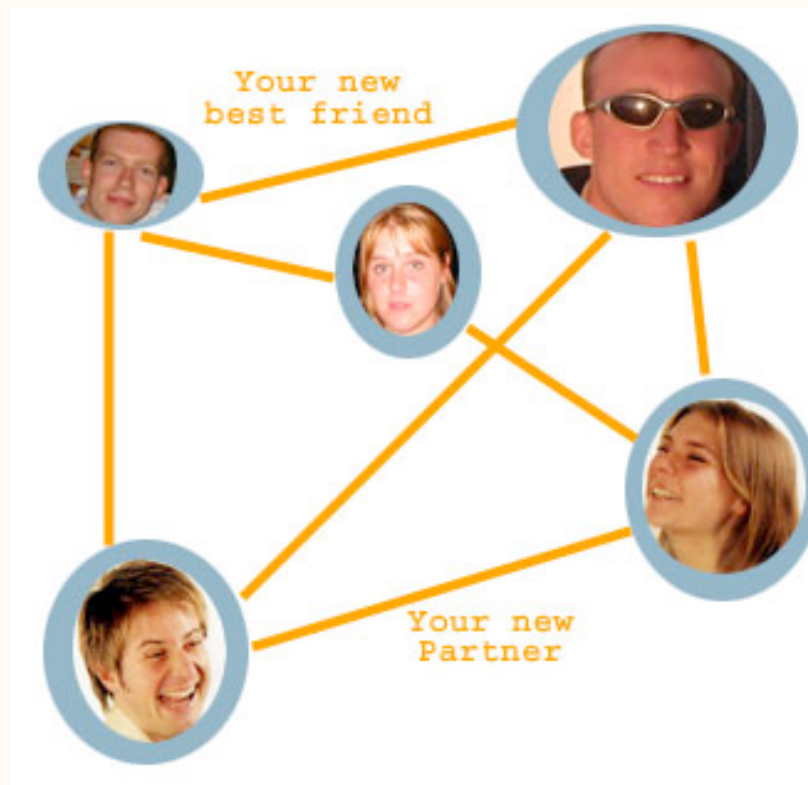


8

Requirements

SLAC Specific Environment

- Little use made so far of Relational Database
 - so make access to SQL v. easy



August 27, 2004



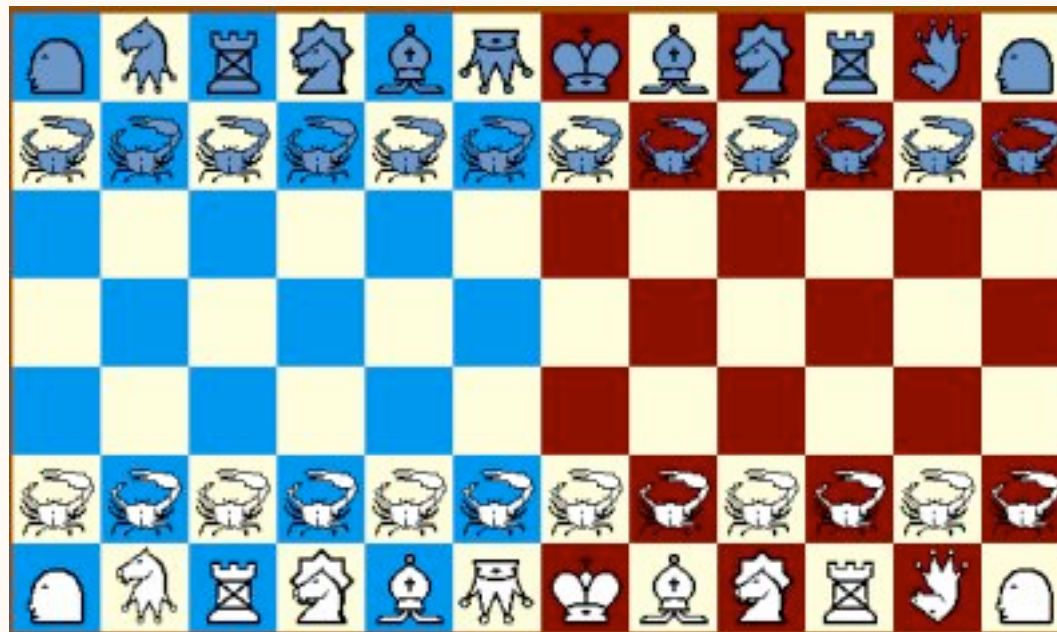
Aida Review



Requirements

SLAC Specific Environment

- Schizophrenic Control System
 - 2 CS Frameworks; 2|3 major O/S; 2|3|4 accelerators?
 - Functionally identical data sources, different API and data



August 27, 2004



Aida Review



10

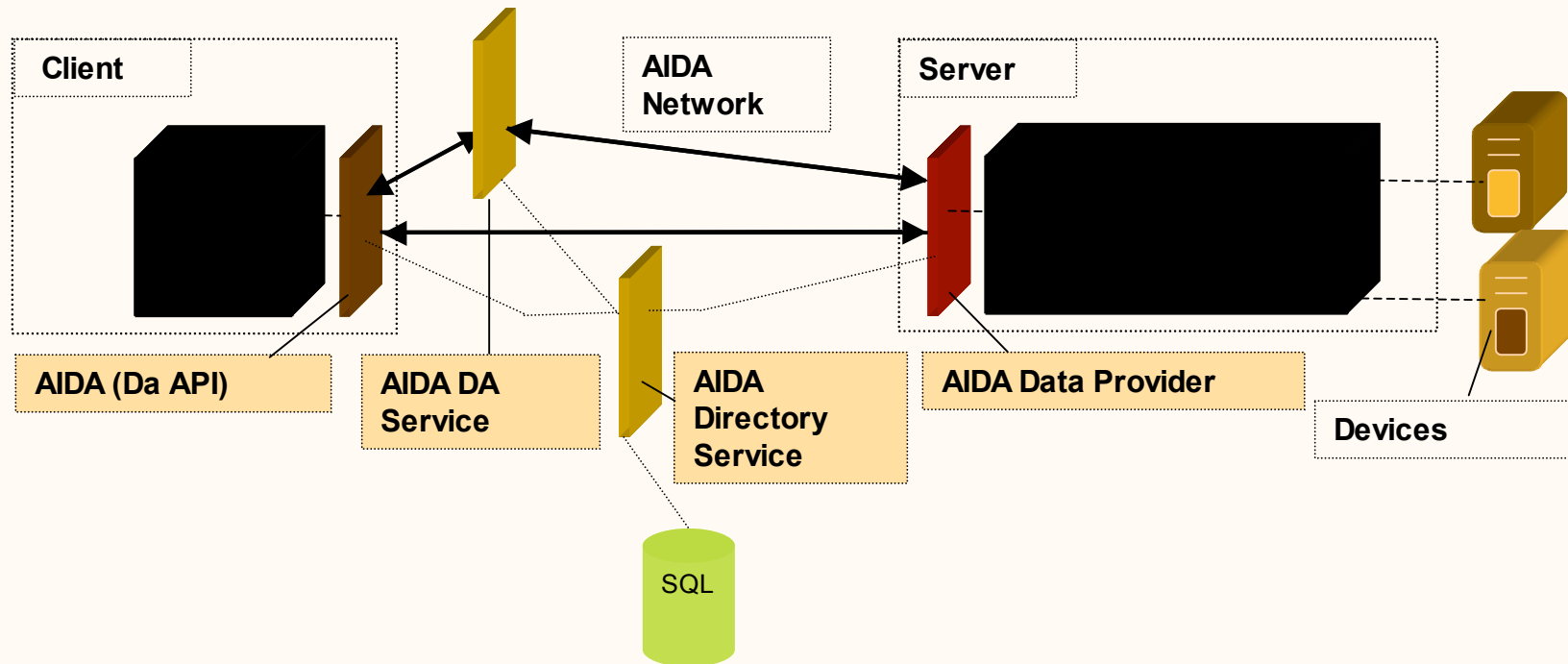
Where Aida fits in

- Wire Scanners, Correlation Plots
- Bdes-to-kmod/kmod-to-bdes
- Orbit Correction, modeling applications
- Ad-hoc offline accelerator physics analysis
- “Optics Plots”
- History plots
- Orbit fitting, BPM plots
- Updating displays
- Control of remote devices
- RT control and feedback



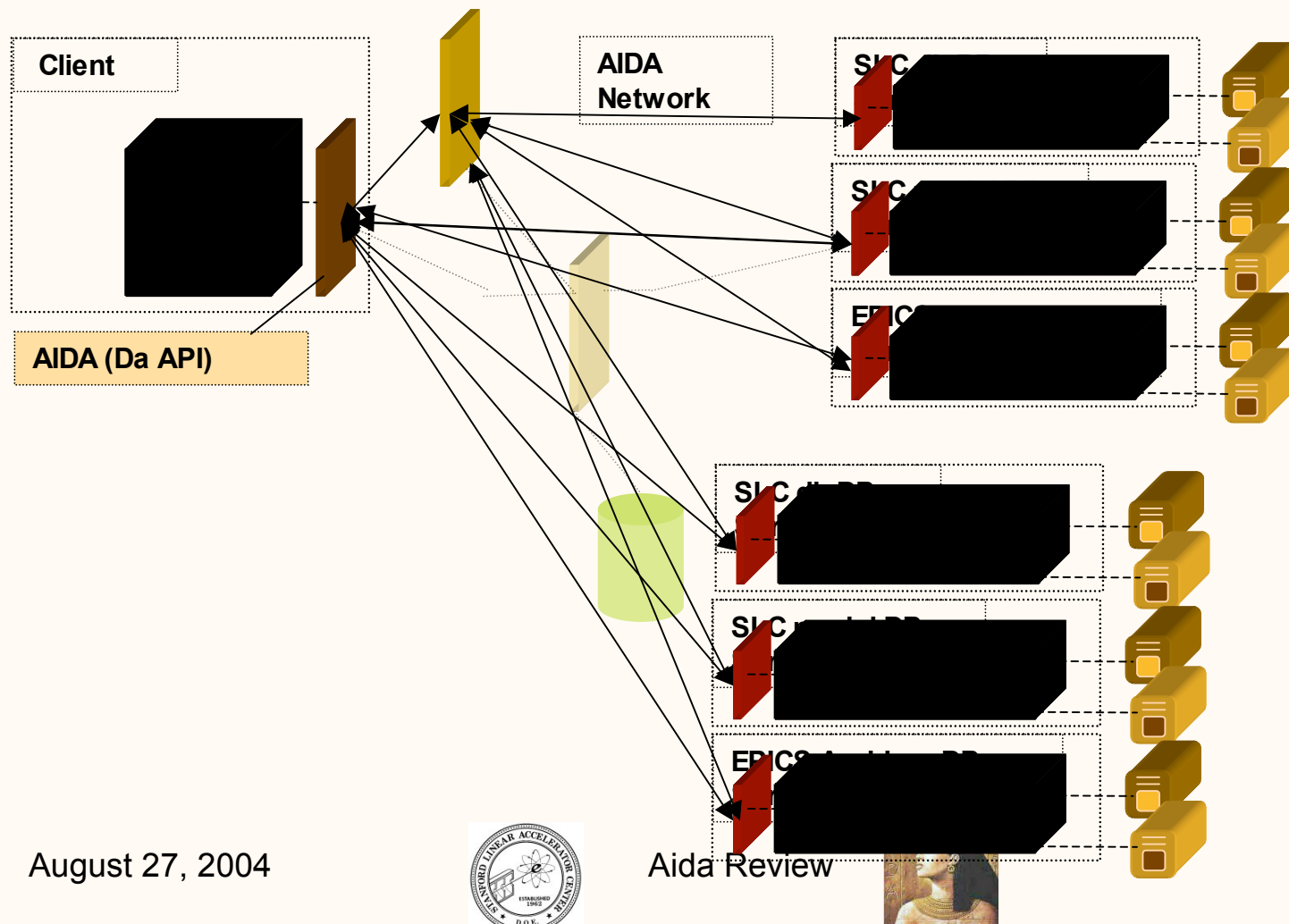
Design

Overall Schematic



Design

Overall Schematic - 1 client : N servers



August 27, 2004

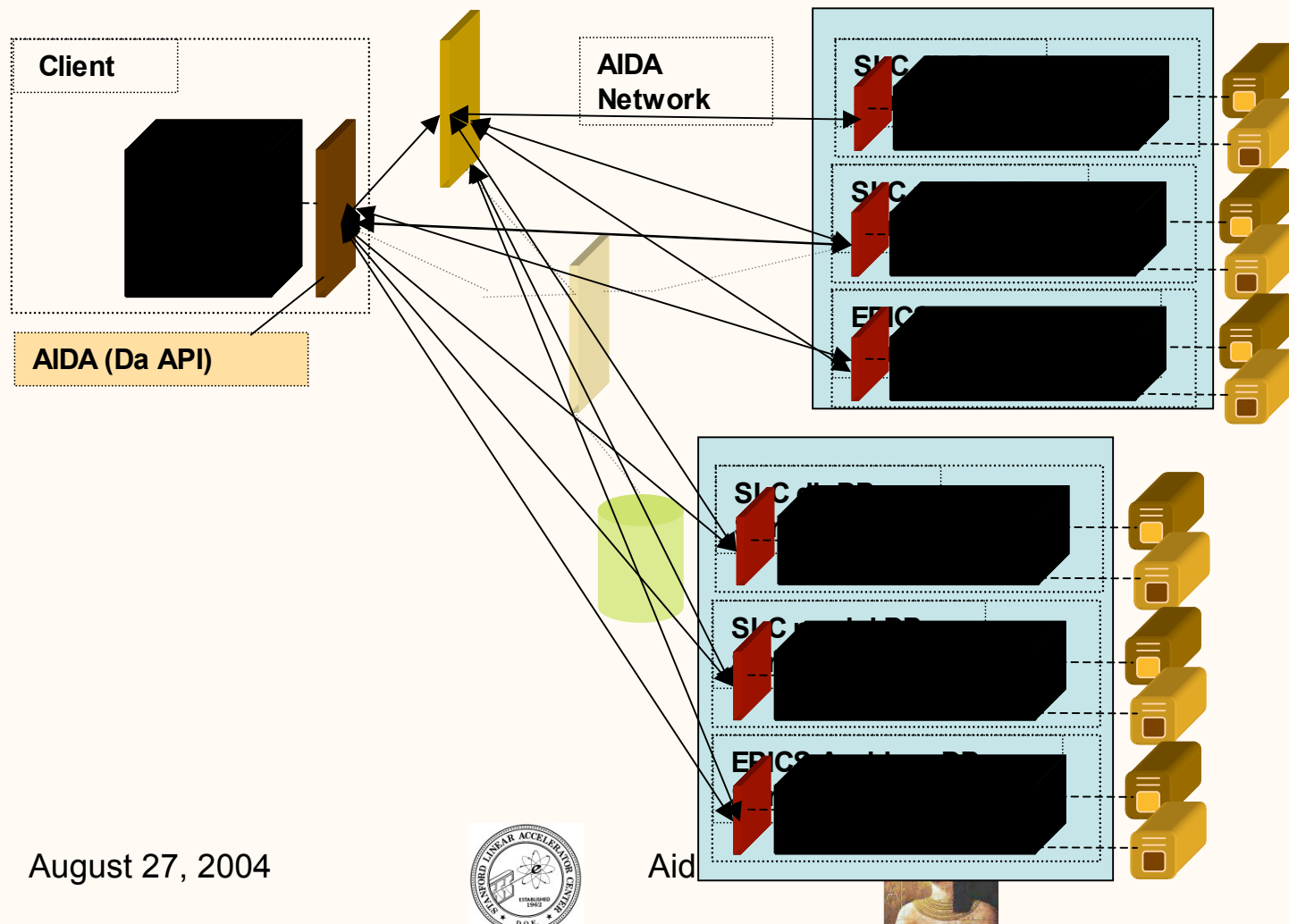


Aida Review

Design

Overall Schematic - Aida "Peers"

Aida peer is a CORBA server

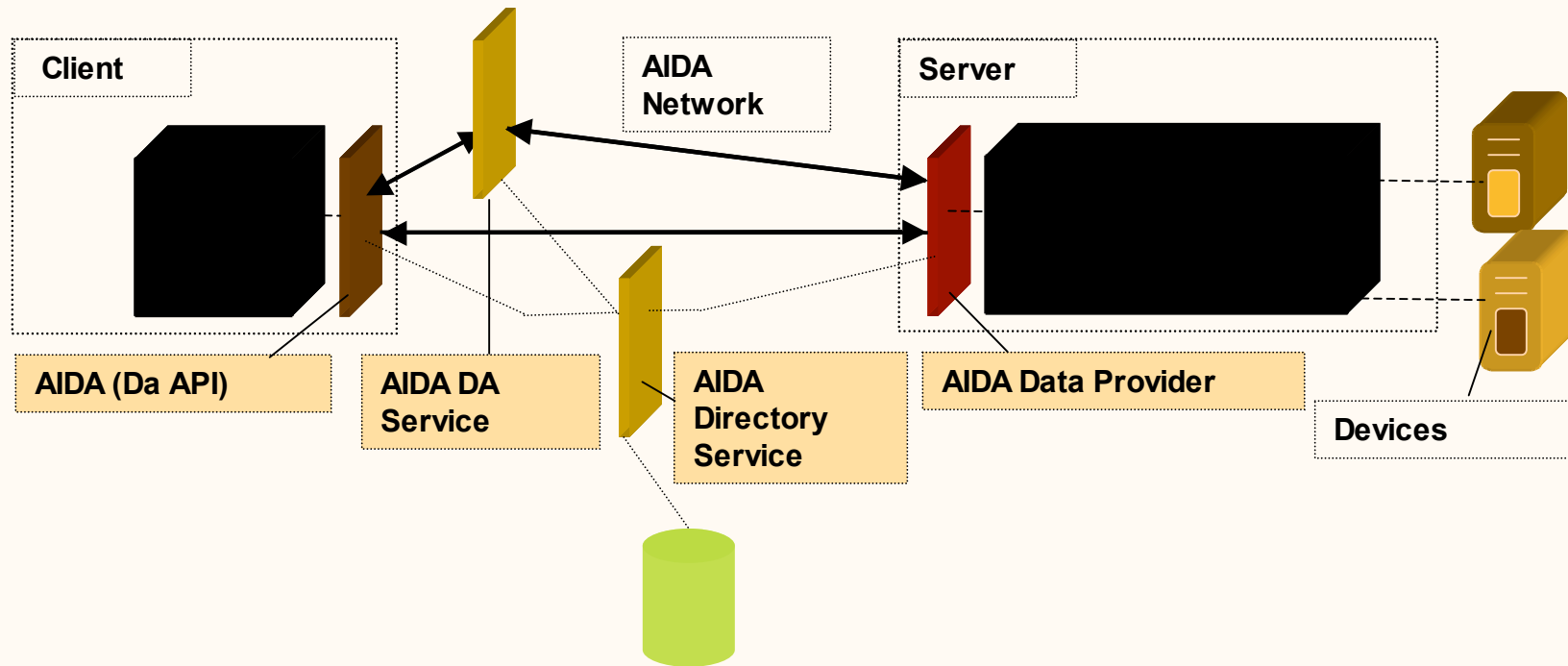


August 27, 2004



Design

Overall Schematic



Status - Done

- The Vision Thing.
- [WBS](#)
- Established CORBA as basic middleware (ORBACUS)
- Core architecture. Peers, Data Providers as “AidaObjects”.
- Data Exchange. So far based on CORBA Any. Not Value-Type.
- Directory Service. Started populating with meta-data.
- Prototyped Data Providers for EPICS ca, EPICS Archivers, SLC db, Test.
- Learnt to manage languages and tools (java, c++, JNI, CORBA; each on Solaris, NT, VMS; netbeans etc)
- Platform independent development environment (java IDE, CVS, make).
Creates Template Data Provider for you.
- Remote server process management. Server Persistence IMR rejected.
- Exception handling out of process, to CMLOG.
- Error handling pattern. Exception definition pattern.
- Prototyped Servlet displaying data to web. Web access of Names Directory
- Prototyped Event based asynchronous exchange (using event service)
- Performance tested alternative data exchange object types, and final architecture.

August 27, 2004



Aida Review



16

Little Demo

August 27, 2004



Aida Review



17

Design Highlights

- Client-server Connection Types
- Aida Directory Service
 - Servant Object Registration and “service groups”
 - The Aida Schema
 - Schema’s implementation in Directory Service Database
- Name translation
- Support for high level “Services”



Design Highlights

Client - Server Connection Types

- Clients can call a service via the DA
 - Can ask the DA to parse and compile queries as well as reconstitute results
- Clients can call the Service directly
- Clients can parse and compile a target reference so that
 - It is validated for correctness and its structure is determined
 - All service references are resolved -> fast data access



Design Highlights

Aida Directory Service

- Directory Service is the key
 - Location found by URL in “slaonly”
 - All AIDA participants Bootstrap by finding the Directory Service first
- Services register with the Directory service when they come on line as authorised handlers of a named set of service requests for a particular Directory Service “group” eg “Dev” or “Prod”
- The Directory service can now respond correctly to all clients in the relevant group (including DA) with the service’s interface



Design Highlights

The Aida “Schema”

- Maps a flat (EPICS) name space to a structured name space.
 - Eg Attribute VAL.HIST or TWISS.X.BETA
 - HB60:DCCT:SUMP.VAL references the EPICS data provider.
 - HB60:DCCT:SUMP.VAL.HIST references the Archiver data provider
 - TWISS.* gets from MODEL data provider
- Visibility of Data Providers and AIDA Servers based on Group
 - Production
 - Development
 - New servers
- Meta Data
 - Optional type and length
 - If not supplied they need to be discovered by the data provider (e.g. history must discover length), but for instance the slc database data provider must be given the length in order to make an acquisition.



Design Highlights

Mapping, Transformations and Services

- Name//attribute
 - Can map to one or more chained names to different data providers:
 - in parallel:
`HB60:DCCT:SUMY//ALL -> HB60:DCCT:SUMY.VAL, HB60:DCCT:SUMY.VAL.HIST`
 - in series (“Services”)
- Transformations
 - Literal:
`HER.I -> HB60:DCCT:SUMY.VAL`
`LINAC.SURVEY.MAG//PED_CNET -> SELECT CENTER FROM MAGS WHERE AREA='LINAC';`
 - or regex: `HB60:DCCT:SUMY.VAL | 's/DCCT/I/g' -> HB60:I:SUMY.VAL`
 - Takes chains of sed like arguments



AIDA Database Schema

1. Look up the instance//attribute combination here. More than one entry may have the same instance//attribute pair

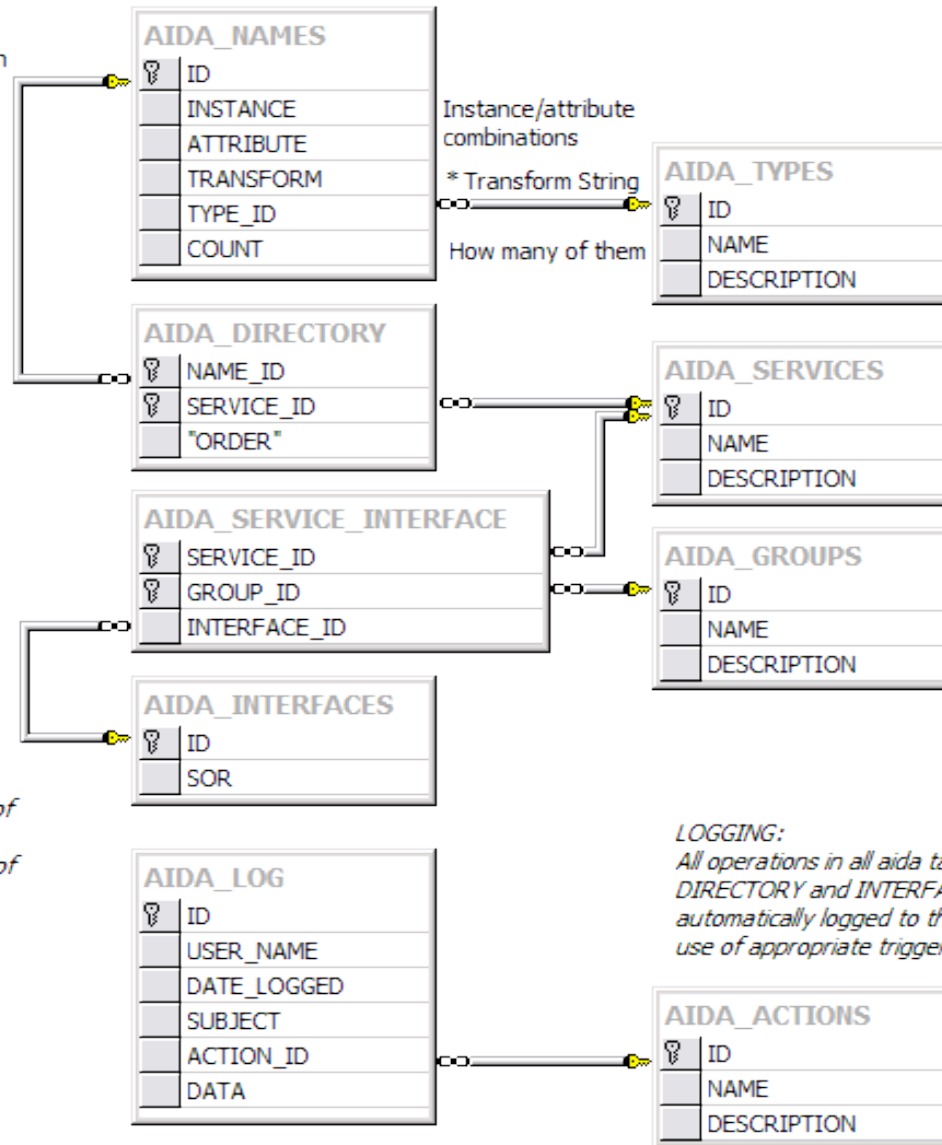
2. Look up the list and order of services that are required to treat this identified request

3. Services register their interface ID's here for each group in which they participate

4. Each interface stores it's string object reference here.

** Transform String*

Null - Simple concatenation of instance//attribute /pattern/replace/[g], ... list of replacement specs
String - any other string is inserted verbatim



LOGGING:

All operations in all aida tables except DIRECTORY and INTERFACES are automatically logged to the aida log table by use of appropriate triggers



Design Highlights

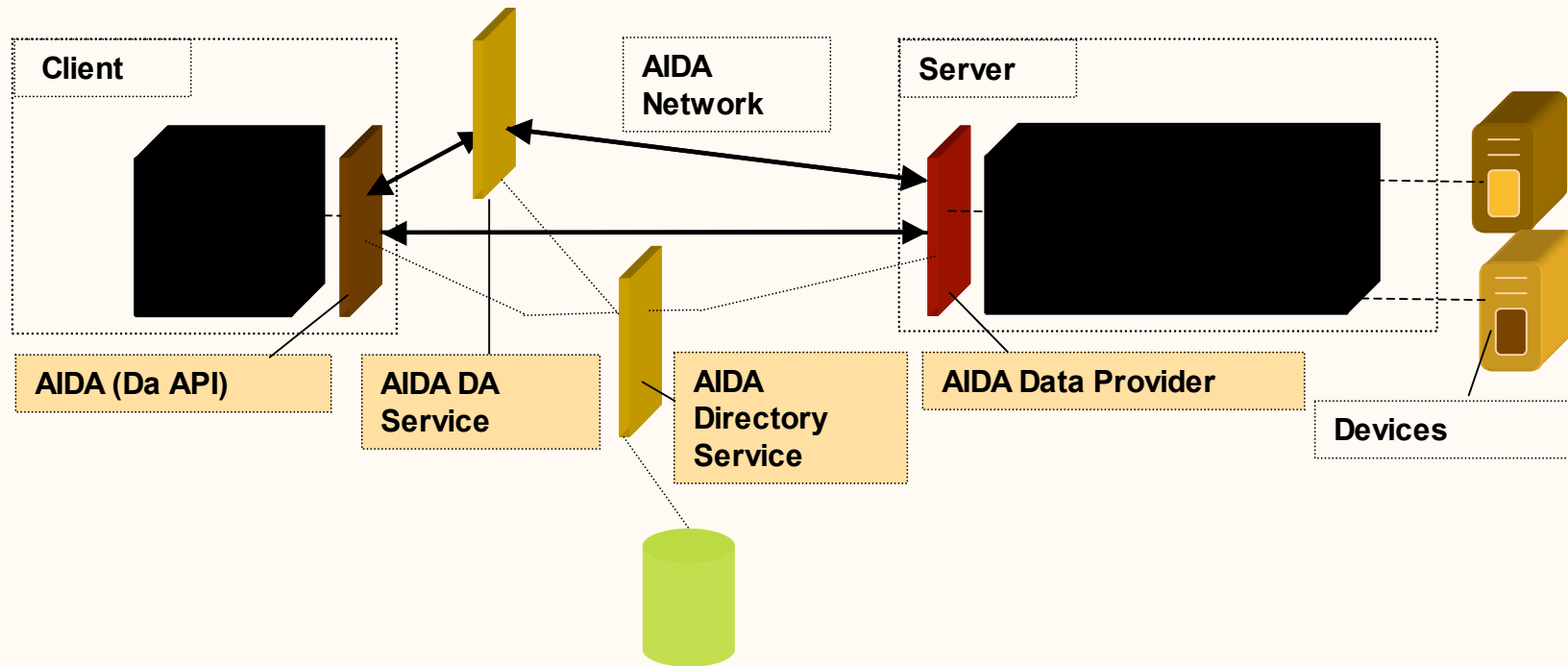
Mapping, Transformations and Services

- Different transforms for each server may be specified
 - Data Provider is called in series "chained"
- An Intermediate Server could manage subordinate servers
 - Eg, Magnet server doing synchronized mag_func().

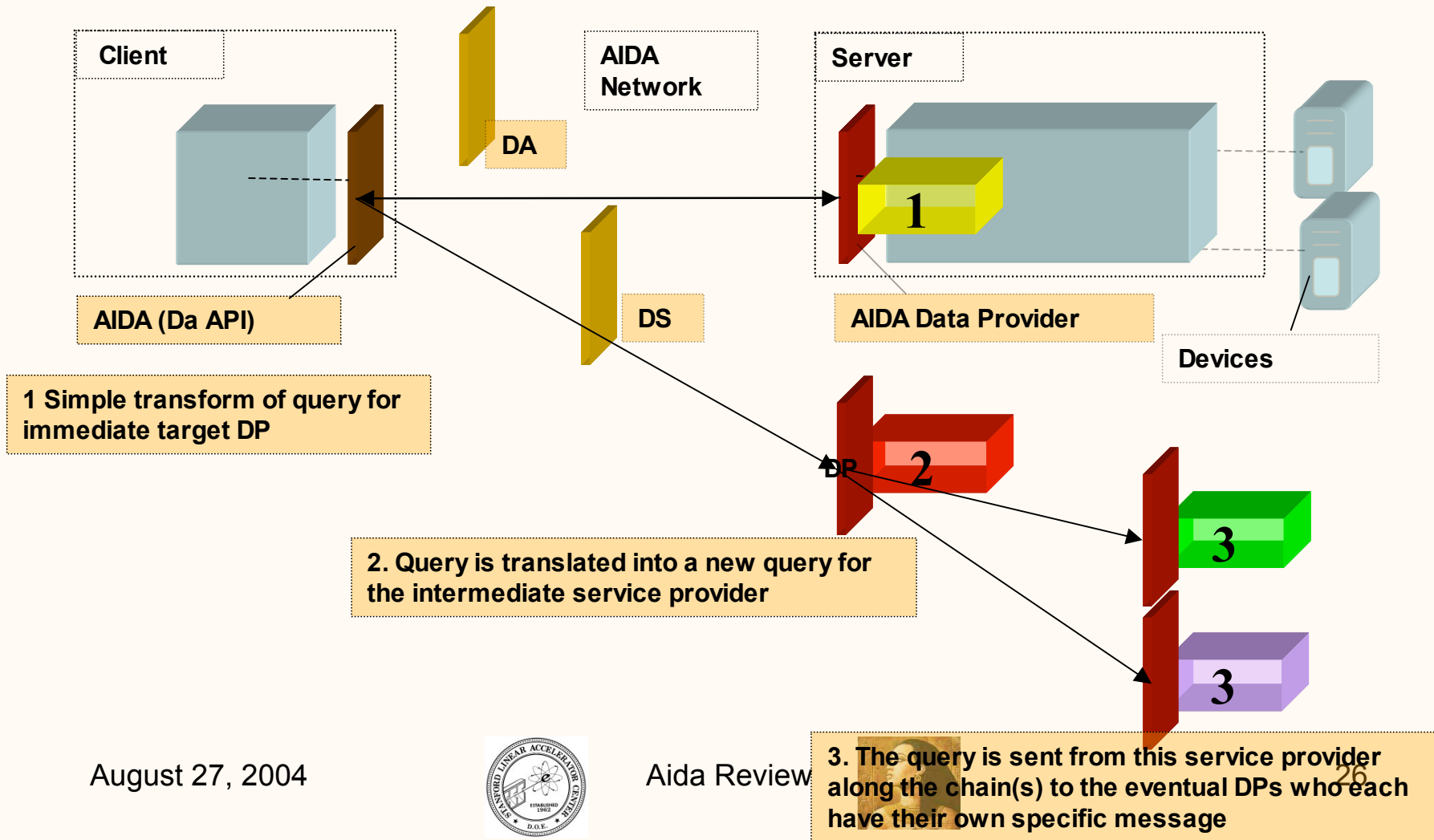


Design

Overall Schematic



Design Services



Management and Administration

- Alternative Technologies
- Prerequisites of Aida
- WBS
- Risks



Management and Administration

Alternative Technologies

- CDEV
 - Unix only: We want VMS, maybe Windows as well
 - Proprietary network protocol vs CORBA well known, continuously developed, not proprietary
 - Inextensible - can't leverage it to complex APIs
 - Only C++, some java binding. Multilanguage, Java, C++, C, based on IDL
 - No Dynamic name mapping, with name translation
 - Flat-file directory service data vs Oracle
 - Flat namespace (1 Entity: N Data Providers, 1 per Attribute) vs n-tier
 - Only EPICS ca and simple model server CDEV DPs existed at the time.
 - No structured data capability
 - No potential for Data Services (DP calls DP)
 - Asynch based on callback, not broadcast push_consumer.
 - Responsibility
- Web Services/GRID
 - Maybe with so-called "fast web services"
 - Grid now oriented towards distributing processing packages, not small data
 - But, some form of Web Services will be ubiquitous!
- Resource Data Framework (RDF)
 - Attractive Future possibility, if done well
 - Name translation
 - Semantic Translation
 - Gives a framework for "services".
- BEEP
 - Absolute-amundo, if wrote our own socket-level protocol.

August 27, 2004



Aida Review



28

Management and Administration

Prerequisites

PRE-PREREQUISITES OF AIDA

- * Process management.
 - Developed IMR management, but chose to develop a system that could be used by other CS processes too.

- * Development Environment.
 - CVS & well known, agreed, file structure
 - managed "reference" area.
 - make, as rest of Unix CS
 - release escalation
 - deployment management
 - Dynamic file protection
 - To Do: remote CVS and sharing with VMS

- * Exception Definition
 - Multi-language and general to rest of CS

- * Error handling
 - Error handling pattern.
 - + exception translation
 - + Logging, throwing, exception translation, all in-line
 - Logging out-of process

- * Monitoring.
 - Not yet started. May use Alarm Handler system of Ron et al, or Yemi's?

August 27, 2004



Aida Review



29

Management and Administration

WBS

August 27, 2004



Aida Review



30

Management and Administration

From Here to “Production”

- Exception Handling
- Solidifying core servers
- Set()
- Security and Authentication
- Production Quality Data Providers
 - SLC History
 - Archiver History
 - SLC Model
 - Oracle
 - SLC db
 - EPICS ca
- Matlab Interface
- Programmers Guide

August 27, 2004



Aida Review



32

Management and Administration

Future

- State, Persistence, Synchronization for services?
- Asynchronous messaging?
- PC & Windows? Servers [Nooww!] or just clients?
- Complex application APIs
 - Eg: Correlation Plots, Wire Scanners, Chromaticity Correction.
 - Require more than just state, need some form of RPC.

