Blaise 5 Data Handling

Arno Rouschen
Blaise Team
Statistics Netherlands

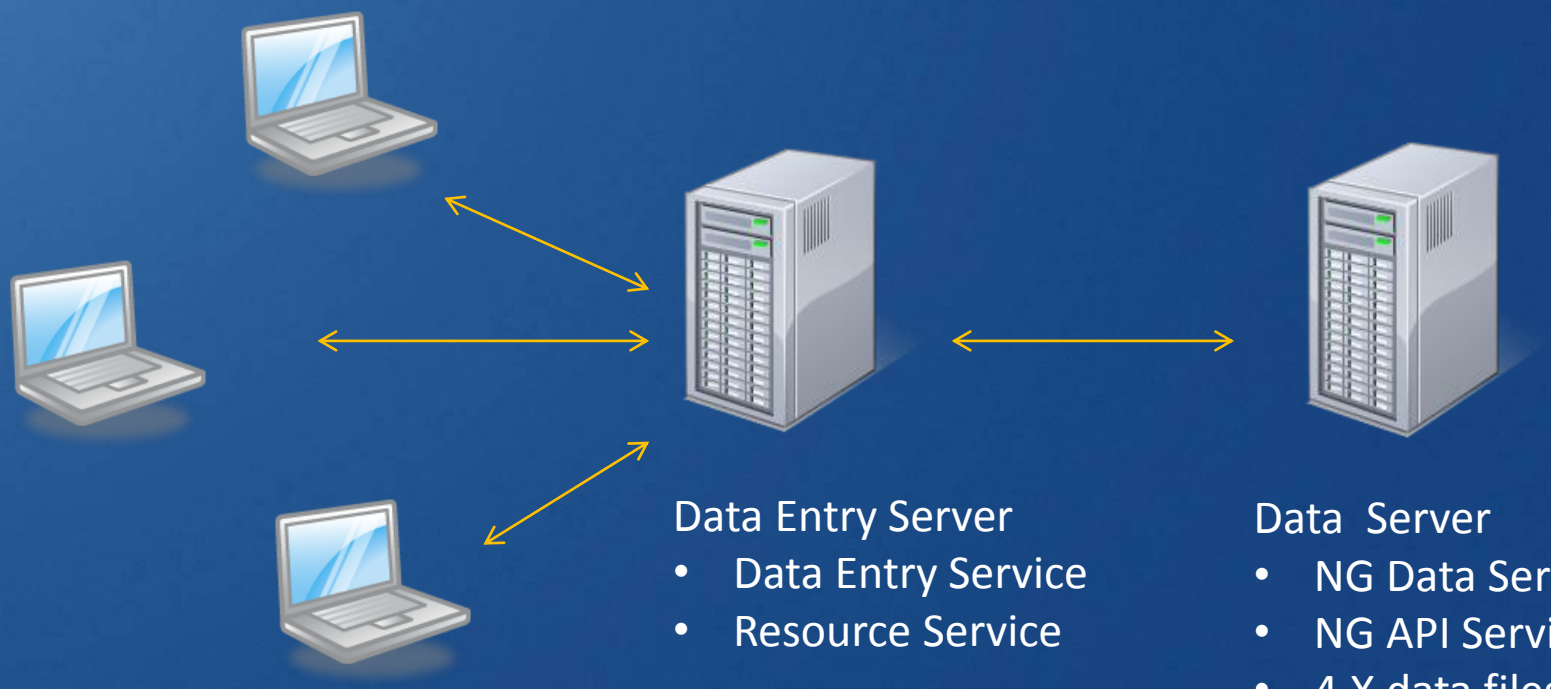
Overview

- Introduction to data handling
- Concepts
  - Data Interfaces
  - Data sources
    - Relational databases, Text files, XML files
  - Blaise databases
- Encryption
- Deployment
- Data access components

Introduction

- Phase 1 (CTP version, released Dec 2010)
  - Blaise NG Data Service
  - Data access via Blaise NG API service
  - Access to Blaise 4.x data files only

Phase 1



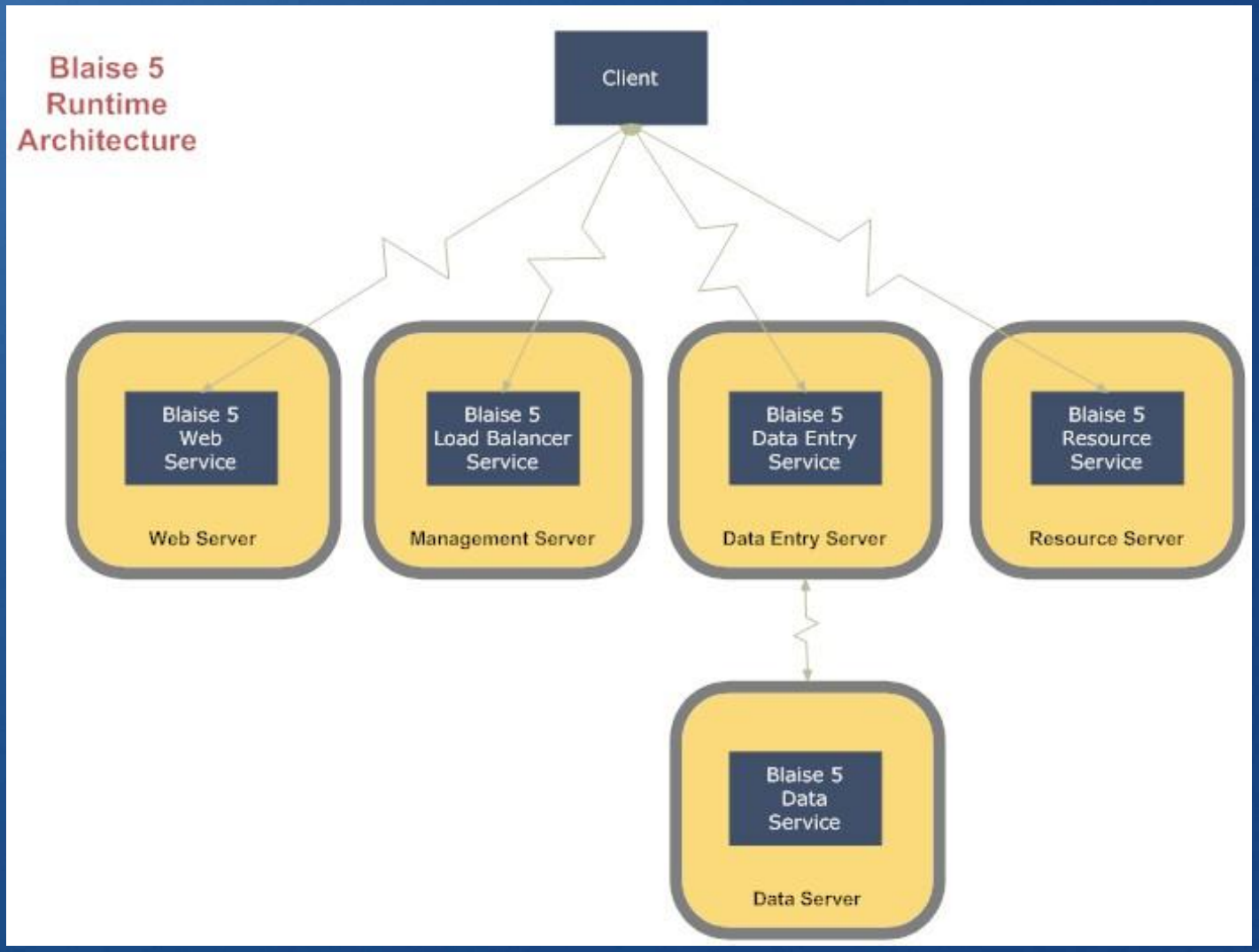
Data Entry Server
- Data Entry Service
- Resource Service

Data Server
- NG Data Service
- NG API Service
- 4.X data files

Introduction

- Phase 2
  - Blaise 5 Data service
  - Data Interface files
  - New Blaise database format
  - Blaise .NET Data Provider

Runtime environment



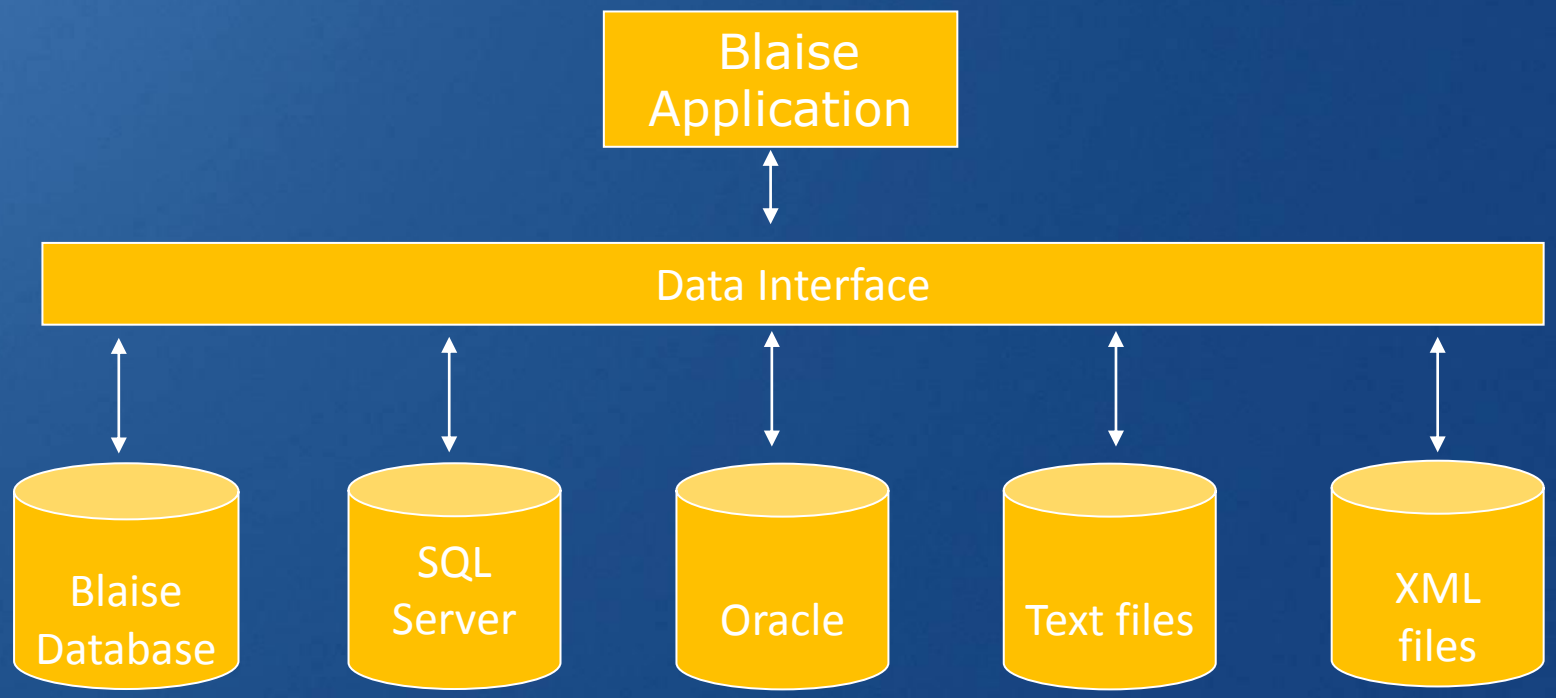
demo

Introduction to Data Interfaces

Data Interface

- Successor of Blaise 4.x boi file
- Main and only data file in Blaise 5 !!!
  - Data access always through data interface
- Can be used to access:
  - Blaise 5 databases
  - Relational databases
  - Text files
  - XML files

Data Interface



Data Interface

- Contains logical definitions; no data
  - Connection Information
  - Table structures
    - Field-Column mappings
  - Associated prepared meta file (*.bmix)
- Stored in a file with *.bdix extension

Data Interface Types

- Based on existing table
  - Mostly used for lookup purposes
  - Contains 1 table
- Fully fledged
  - Used to collect questionnaire data
  - Contains multiple tables
- Reference to another data file
  - Used to access text and xml files

Supported databases

- Blaise 5 Databases
- Oracle, Microsoft SQL Server
  - Access via .NET Framework Data Providers
- Future versions will support MySQL, DB2, Microsoft Access, Sybase, …

Blaise 5 Database (*.bdbx)

- Uses SQLite as storage platform
  - Lightweight SQL database
  - Open source; C++ code is available
  - Used in high profile projects; Safari and Firefox browsers; electronic devices; Adobe software, ...
  - Transactional, zero-administration, serverless, self-contained
  - Unicode support; Multi-platform
  - 32 bits and 64 bits
  - SQLite .NET Data Provider

Database comparison

Blaise 4.x database

- File based proprietary database
  - Stores data as binary stream
  - No versioning
- ANSI
- Non-transactional
- No built-in encryption
- Windows platform

- Data access requires dedicated implementation

Blaise 5 database

- File based relational database
  - Supports multiple table structures
  - Versioning (optional)
- Unicode
- Transactional
- Encryption is possible
- Multiple platforms

- Uses common data access implementation

BOI / BDIX comparison

Blaise 4.x boi file

- OLE DB connections
- ODBC connections
- Other data files
  - BDB
  - BOI
  - Text files
- Generic / non-generic
- Versioning (only generic)
- 5 data partition types

Blaise 5 bdix file

- ADO.NET connections
- OLE DB connections
- ODBC connections
- Other data files
  - BDIX
  - Text files
  - XML files
- Generic / non-generic
- Versioning (optionally)
- 4 data partition types

Using an existing table

- Steps
  - Create Data Interface
  - Connect to database
  - Select table
  - Generate Blaise meta for this table
  - Save and use data interface

demo

Creating a Data Interface file for an existing table

Storing questionnaire data

- Needs a fully fledged data interface
- Create options:
  - Implicitly by clicking Run button in Control Centre
    - Creates default data interface and Blaise database
  - Explicitly by using a wizard
    - Choose database platform
    - Choose the structure of your data tables
    - Choose between generic and non-generic tables
    - Choose whether to add versioning columns

Storing questionnaire data

- Steps
  - Create data model and prepare it
  - Create data interface
    - Specify connection to target database
    - Select a prepared data model
    - Choose table structure
    - Create tables in database
    - Save data interface
  - Use data interface

demo

Creating a Data Interface file in order to store questionnaire data

Using text files

- Steps
  - Create and prepare data model
  - Create data interface
    - Specify connection parameters
      - Path to text file
      - Text file attributes
        - Field separator
        - String Delimiter
        - Date and time format
  - Save and use data interface

demo

Using text files

Attach to Blaise 4

- Blaise XML files are readable by both versions
- Steps
  - Create data interface for XML file
  - Export Blaise 5 data to XML
  - Use Blaise 4.X tools to process your data

demo

Using Blaise 4 data in Blaise 5 and vice versa

Data Security

- By securing your data server
  - Data service process is the only one who needs access to the data interface files
- By using Blaise features
  - Protect your data interface file
  - Encrypt your Blaise database file

Data Interface protection

- Connection string
  - Might contain confidential information
  - Will always be stored encrypted on disk
  - But, by default, will be exposed whenever you load the file from disk
  - Can be protected by providing a password

demo

Protecting your data interface

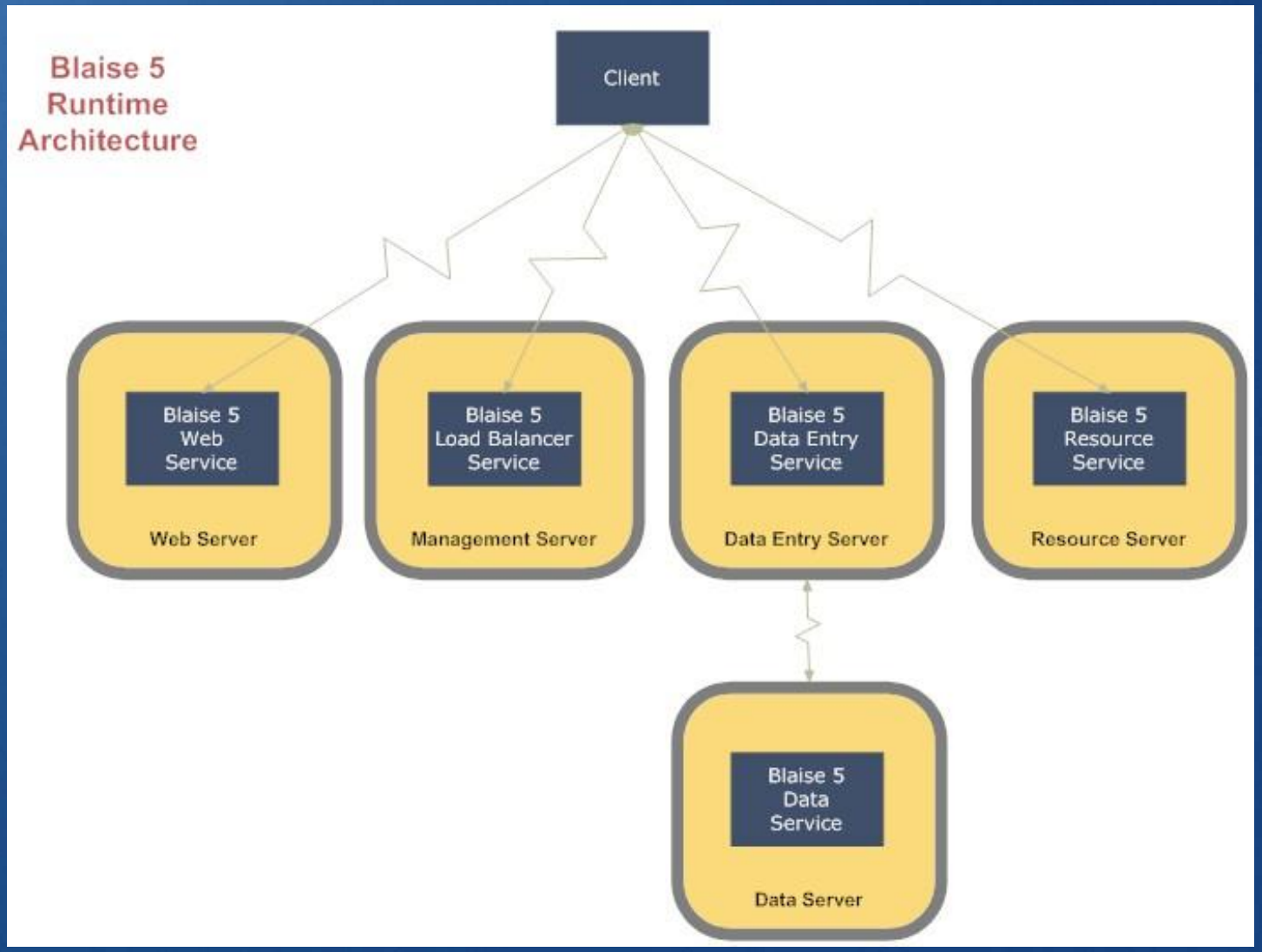
Database file encryption

- SQLiteCrypt module
- Encrypts data on the storage layer
- Uses AES encryption algorithm
  - Approved by US Government
  - Very good performance
- Multi platform
- Zero configuration
  - No setup or administration needed
  - No key management
- We get it for free and is very easy to use!!

demo

Encrypting your database file

Deployment


Blaise 5
Runtime
Architecture
Client
Blaise 5
Web
Service
Web Server
Blaise 5
Load Balancer
Service
Management Server
Blaise 5
Data Entry
Service
Data Entry Server
Blaise 5
Resource
Service
Resource Server
Blaise 5
Data
Service
Data Server

Deployment

- Blaise Data Server
  - There can be only one
  - Stores data interface and database files
  - Runs Blaise 5 Data Service
    - Invokes Blaise .NET Data Provider
      - Accesses data sources

Deployment

- Project doesn’t contain a data interface
  - Default data interface and default Blaise database will be created and packaged
- Project contains a data interface
  - Data interface will be included in package
  - Local database files will be included in package

Deployment

- Local databases
  - Blaise databases, Microsoft Access databases
  - Connection string should contain relative path to database file
  - Database file will be included in deployment package and stored on data server
- Server databases
  - SQL Server, Oracle, MySQL, DB2
  - Data Interface will be deployed to data server
    - Connection string must be valid according data server

Data Access Components

Client objects

- DataRecord
- BlaiseDataSet
- DataInterface

Blaise .NET Data Provider

- BlaiseConnection
- BlaiseCommand
- BlaiseDataReader
- BlaiseDataAdapter
- SQLComposer (Used Internally)

Blaise SQL Parser

- Builds Abstract Syntax Tree for a Blaise SQL statement

Data Access Layer

- Classes for Abstract Data Access
- Object Oriented SQL Statement Builder

Blaise .NET Data Provider

- Used by Blaise to access data sources
- Implements obligatory .NET Data Provider interfaces
- Can be used in order to access
  - Blaise database files
  - Relational databases
  - Text and XML files via Data Interface
- ‘Talks’ Blaise SQL
  - SQL using Blaise syntax rules and reserved words

BDP: Mandatory objects

- BlaiseConnection
  - Opens connection to data source
- BlaiseCommand
  - Executes command against the data souce
- BlaiseDataReader
  - Fast forward read-only cursor
- BlaiseDataAdapter
  - Fills and updates data sets

Blaise SQL Parser

- ANTLR based
- Uses a parser and lexer which are created via Blaise SQL grammar file
- Generates statement dependent AST for
  - Select statements
  - Insert statements
  - Delete statements
  - Update statements
  - …

Data Access Layer (DAL)

- Interacts with the data source
- Object oriented approach for building SQL statements
- Builds data sources native SQL

Flow example