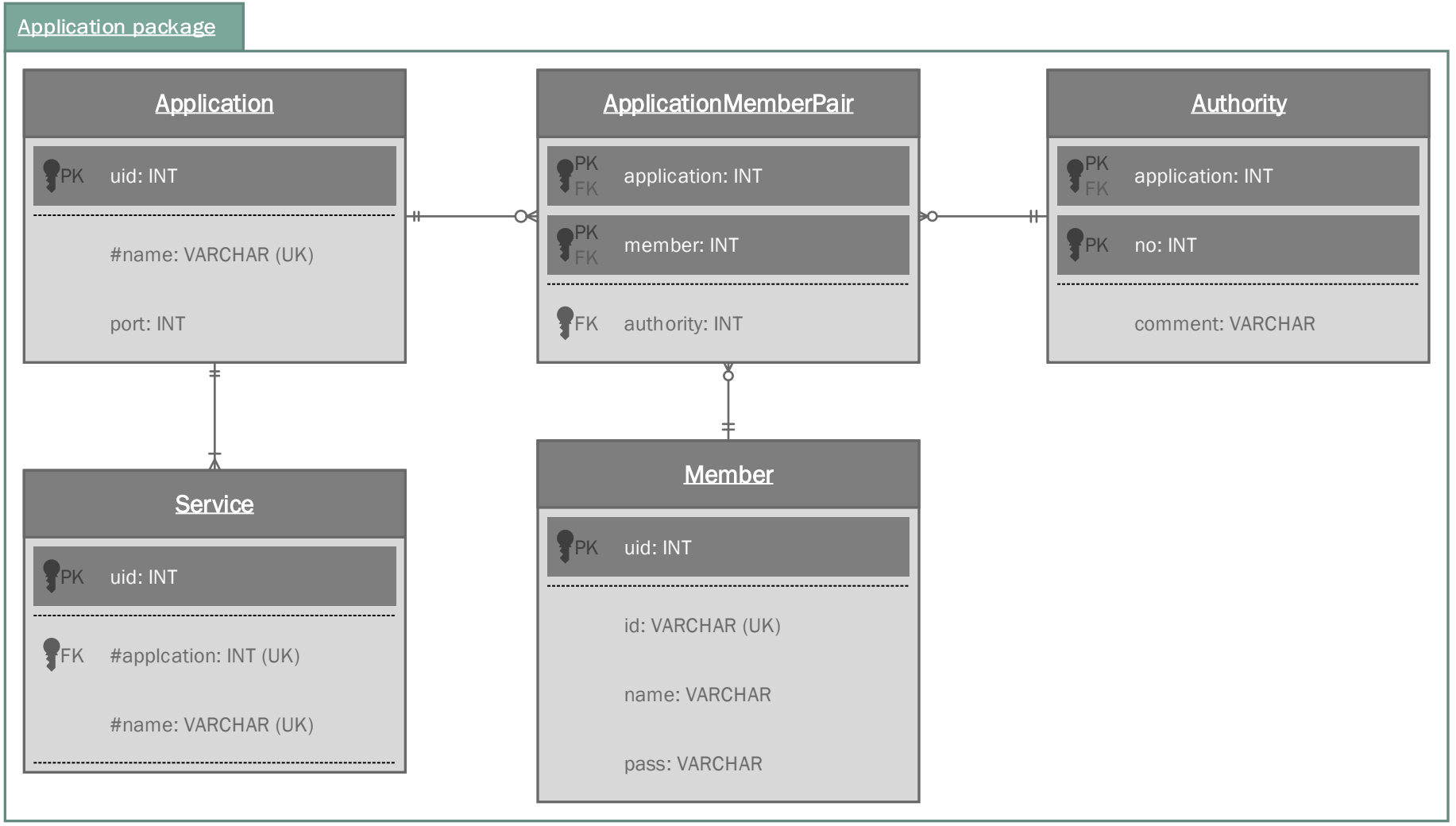
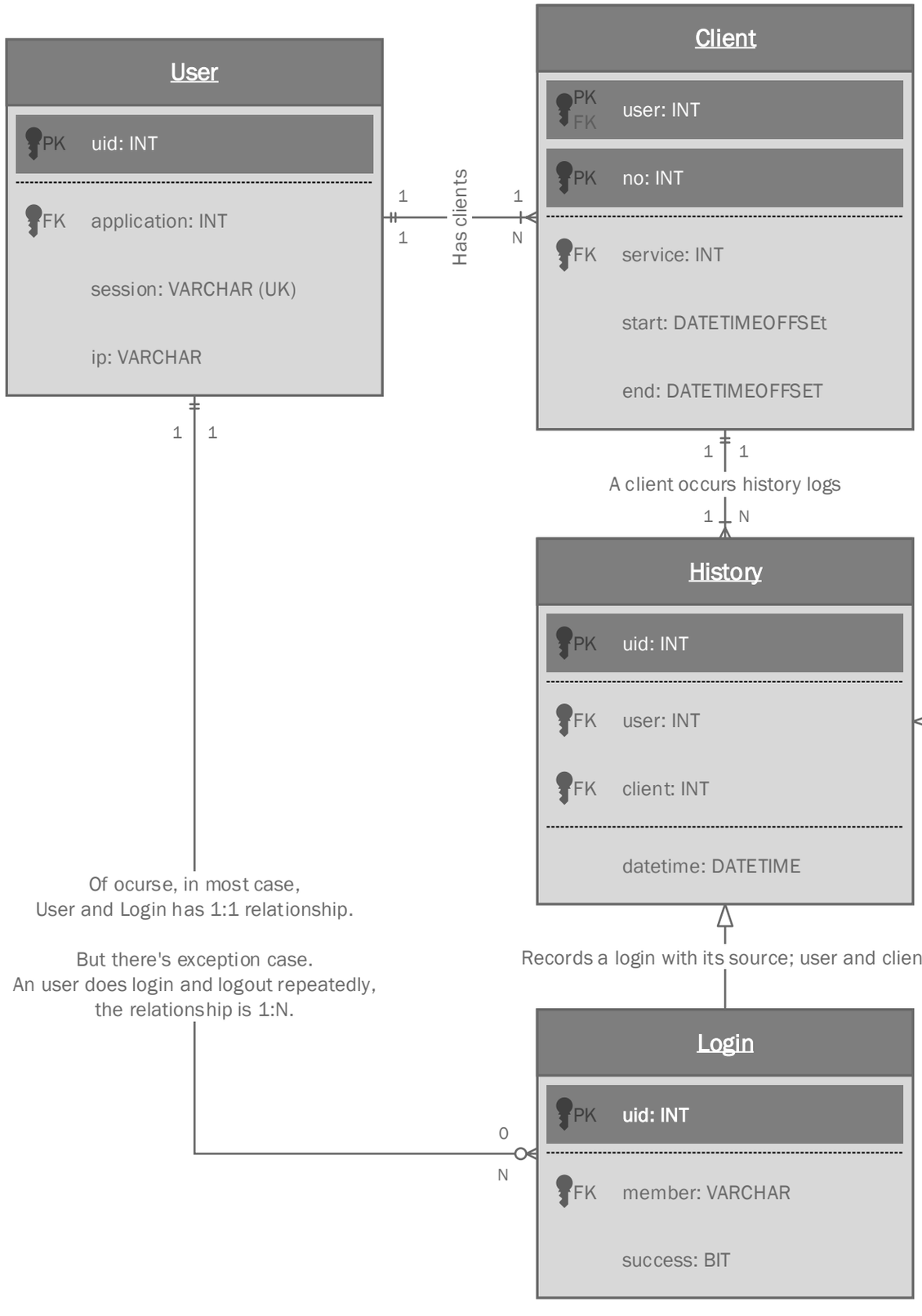
Entity Relationship Diagram

1. Application
2. History Log
3. External System
4. File & Nam-Tree


Application package
Application
PK uid: INT
#name: VARCHAR (UK)
port: INT
ApplicationMemberPair
PK FK application: INT
PK FK member: INT
FK authority: INT
Authority
PK FK application: INT
PK no: INT
comment: VARCHAR
Service
PK uid: INT
FK #applcation: INT (UK)
#name: VARCHAR (UK)
Member
PK uid: INT
id: VARCHAR (UK)
name: VARCHAR
pass: VARCHAR

Historical Log of Service Package



User

application: enumeration code of the cloud
session: session id allocated to an User

Represents an user containing multiple clients which is identified a session id issued from the C++ cloud server.

Client

service: enumeration code of matched service

Represents a physical client running on a browser (JS) or flash player (Flex). Sometimes the physical client can be C# or another languages network driver.
It contains a service and history records like **Login** or **Invoke**

History

Sub-type of all history records

Login

To identify an user, its account
An user, identified by its seeision ID, can have multiple login and logout records and multiply authorization with multiple accounts

Invoke

Represents which command has come from a (physical) client like JS or Flex.
Archiving **InvokeParameter**, the parameters of an Invoke, it's not compulsory. You can omit archiving those parameters for saving disk space of DB

Error

Means whether an error has occurred by an Invoke message from physical client.
The error message means std::exception's message occurred in C++ cloud server.

InvokeParameter

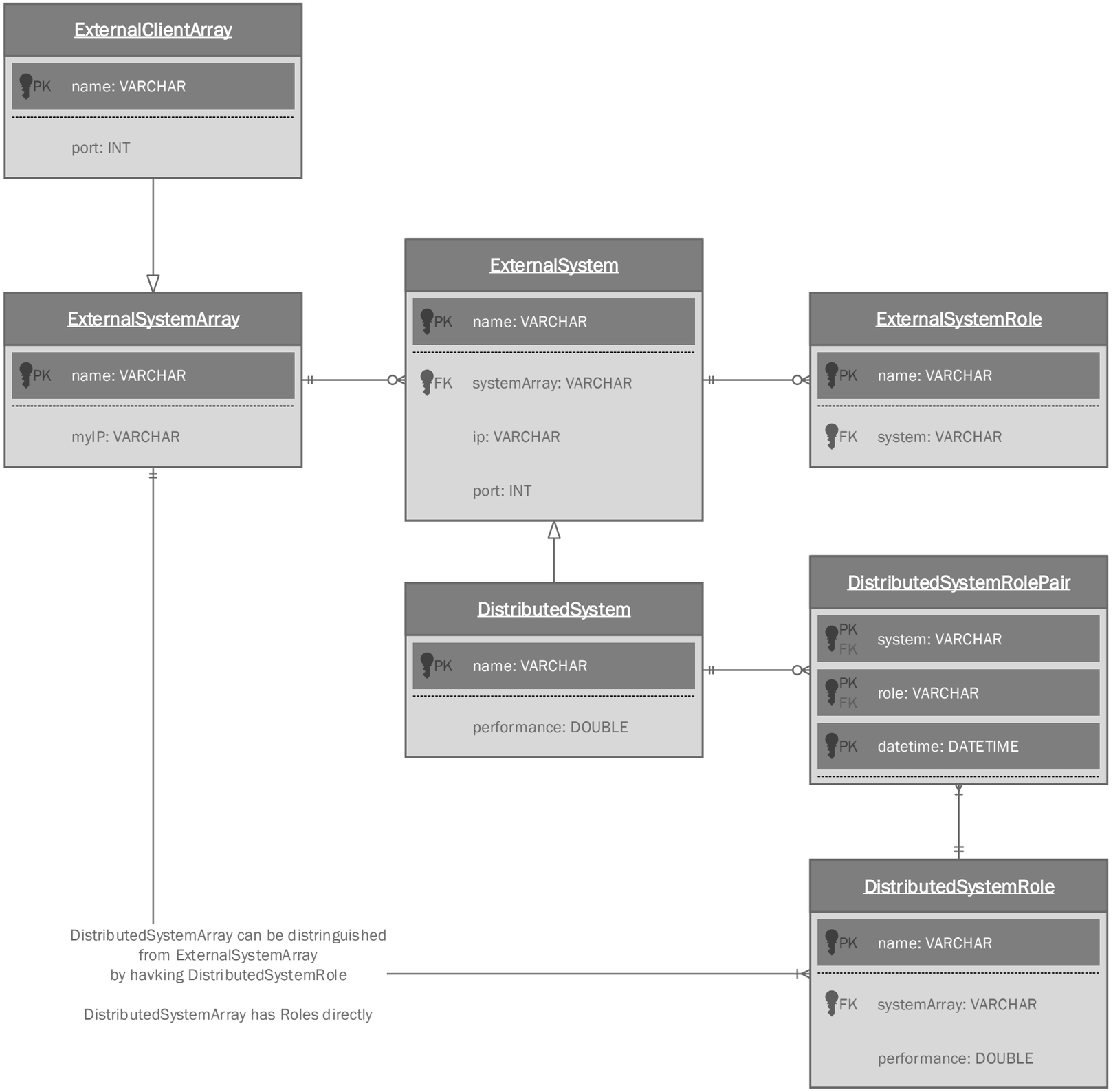
If type of an InvokeParameter is not string or number

If the type is XML, archives a string represents the XML object. (XML::toString()).
Else if the value type is ByteArray, archives its size only. The rule of handling ByteArray is also adjusted to the case of type Base64

Omit InvokeParameter(s)

You can omit InvokeParameter(s) belongs to Invoke or omit value of a InvokeParameter to avoid wasting disk space.
But one thing you've to know is, the Invoke and InvokeParameter tables are used to archiving history logs and monitorig service. The omission will reduce quality of those purposes.

External System Package



ExternalSystemArray

name: A name can represents and ensure uniqueness
myIP: An specialized IP address of my system.

How to distinguish

ExternalServerArray
ExternalSystemArray
Doesn't need any specialized table.
ExternalClientArray
ExternalSystemArray + ExternalClientArray

ExtenalSystem

name: A name can represents and ensure uniqueness
ip: IP address of the external system
port: Port number of the external system

How to Distinguish

ExternalServer
systemArray does not exist in ExternalClientArray
ExternalClient
systemArray exists in ExternalClientArray

DistributedSystem

Performance: A performance index.
The performance index changes and evolves elapsed time of handling requested processes.

How to Distinguish

DistributedSystemArray from ExternalSystemArray
i) Whether has DistributedSystemRole(s) or not.
Ii) Whether children system is DistributedSystem or not.

ParallelSystemArray from DistributedSystemArray
Children DistributedSystem(s) are found, but any related record of DistributedSystemRole is not found.

ParallelSystem from DistributedSystem
Cannot find any related record of DistributedSystemRole

DistributedSystemRole

name A name can represent a role of a distributed processing system
Performance: An index of required performance.
The index changes and evoloves by elapsed time of handling requested processes which are defined in the role

DistributedSystemRolePair

datetime: Time of a role allocation to a distributed processing system.
The datetime even can be used to identify which distributed system(DistributedClient)s are connected to a master(DistributedClientArray).

Nam-Tree Package

File Package

Folder
- PK uid: INT
- FK fid: INT (UK)
- name: VARCHAR (UK)
- FK application: INT
- FK category: INT
- FK memberID: VARCHAR

File
- PK FK uid: INT
- extension: VARCHAR

BinaryFile
- PK FK uid: INT
- data: VARBINARY(MAX)

TextFile
- PK FK uid: INT
- content:VARCHAR

Nam-Tree File Package

NTFile
- PK uid: INT
- header: VARCHAR
- body: VARCHAR

NTExploreParameter
- minimum: DOUBLE
- maximum: DOUBLE
- bin: DOUBLE

NTParameter
- PK FK fileUID: INT
- PK no: INT
- name: VARCHAR
- initialValue: DOUBLE

NTParameterDetermined
- PK FK fileUID: INT
- PK FK parameterNo: INT
- PK no: INT
- label: VARCHAR
- value: DOUBLE

Nam-Tree Criteria Package

NTCriteriaFile
- PK FK uid: INT

NTCriteria
- PK uid: INT
- FK fid: INT
- FK fileUID: INT
- type: INT

NTSide
- PK FK criteriaUID: INT
- PK direction: INT
- FK fileUID: INT

NTSideParameter
- PK FK criteriaUID: INT
- PK FK sideDirection: INT
- PK name: VARCHAR
- value: DOUBLE