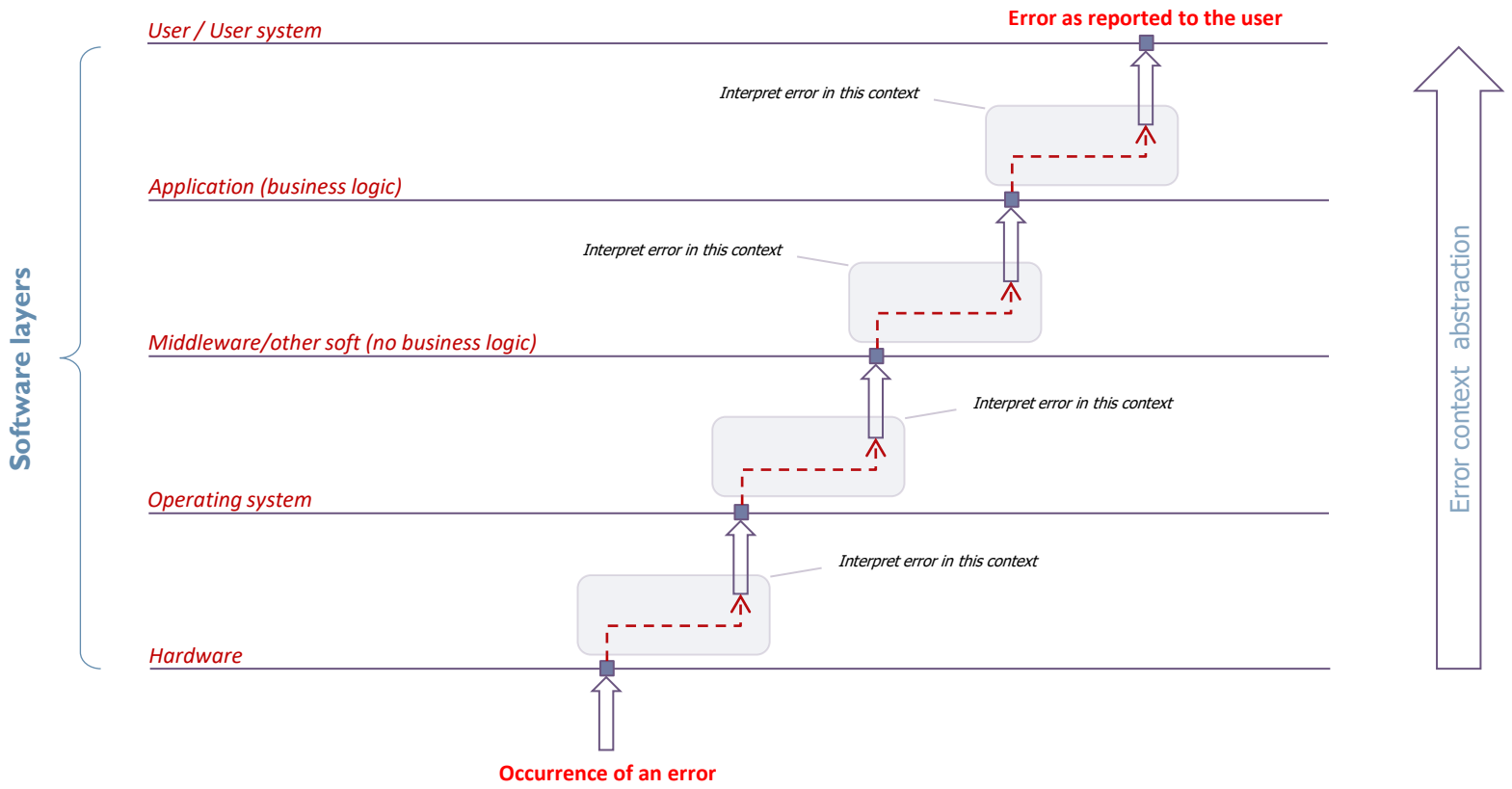
Error reporting logic

And software abstraction levels

Table of content

Software layers and error contexts



Error abstraction logic

- An error originating from a "lower" layer must be interpreted in the receiving layer context before transmitted to the "higher" layer(s).
- The business user (or system) must receive an error meaningful in his (its) business context
- In case of a user, the error must be human readable
- The cascade of error interpretation must be navigable
  - Each layer error interpretation are linked together (navigable both ways)

Error abstraction example

- Hardware : A physical disk cannot be accessed
- OS : The file <filename> cannot be accessed
- Middleware (e.g. SOA) : The Contact Management service cannot be accessed
- Application : Impossible to update Mister X Contact infos.

There is no point in telling the final user that a physical disk cannot be accessed.

Error attributes I

The following minimum set of error attributes is mandatory:

- A time stamp
  - Universal time if possible to allow reconciliation of errors originating from different technological stacks/systems (this supposes time synchronization between systems: may be legally required/binding)
- A unique error identifier
  - A globally unique one if possible (GUID)
  - Allows processing of the error by a system/application (automation)
- An origin reference
  - The component/module/function … name triggering the error
  - A component/module/function origin identifier

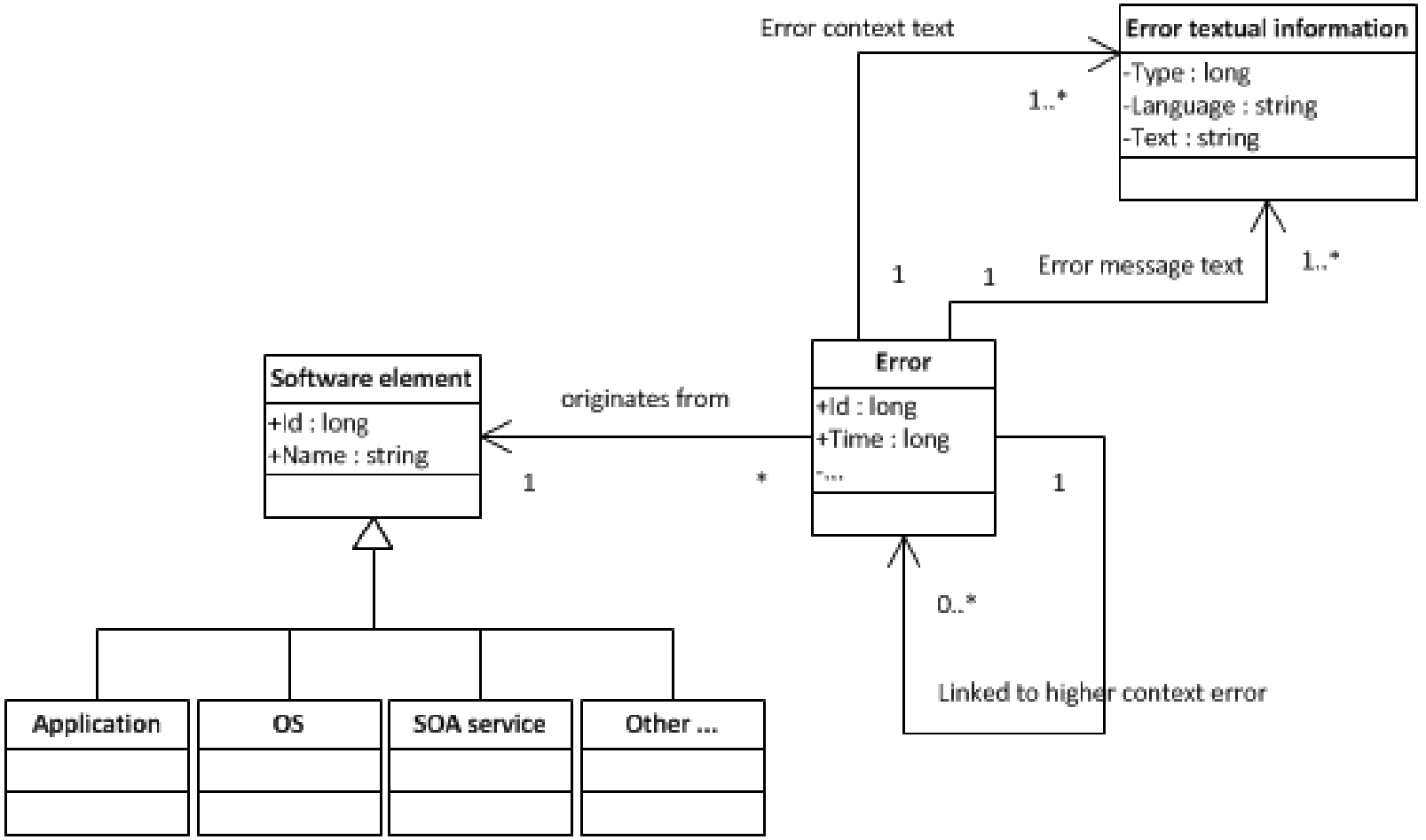
Error attributes II (continued)

- A contextual information
  - The action that was attempted or …
  - The current logical context (e.g. in case of an event)
  - Human Readable
- An error message
  - As unambiguous as possible (e.g. no "an unknown error has occurred")
  - Human readable
  - Multiple languages must be supported (at business level)

Multi-language support

- Multiple languages must be supported for errors reporting to the business layer(s)
- Other software layers (non business) must all support one and unique coding language e.g. English
  - This relates to software engineering conventions and coding styles (name of variables and other software objects, database table and field names, etc)

Error logical data model



Error software development kit (SDK)

- Must be made available on each technological stacks
- Implements the same error handling logic (when applicable)

End of presentation

Questions