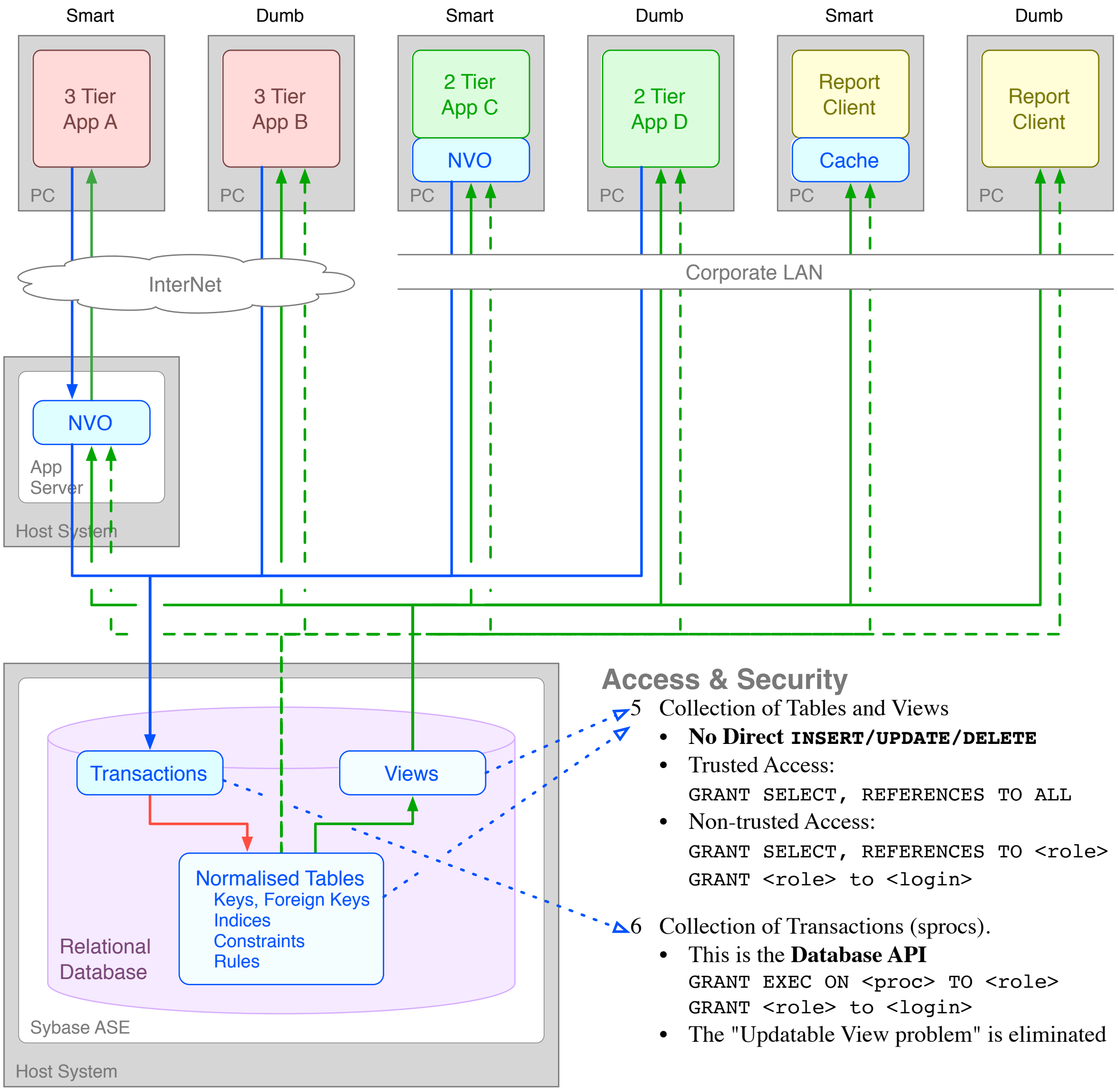
Three Tier Architecture

Open Architecture



- This document defines the facilities that must be implemented in order to comply with **Open Architecture Standards**.
- **Architecture** consists of the proper separation of functional components, thence their deployment, based on the platforms that are available. That is, Normalisation of the software componentry.

Access & Security

5 Collection of Tables and Views

- **No Direct `INSERT/UPDATE/DELETE`**
- Trusted Access:
  `GRANT SELECT, REFERENCES TO ALL`
- Non-trusted Access:
  `GRANT SELECT, REFERENCES TO <role>`
  `GRANT <role> to <login>`

6 Collection of Transactions (sprocs).

- This is the **Database API**
  `GRANT EXEC ON <proc> TO <role>`
  `GRANT <role> to <login>`
- The "Updatable View problem" is eliminated

1 Use `<role/login>` over `<group/user>`
2 All `GRANTS` to `<role>` only
No direct `GRANTS` to `<login>/<user>`
3 Maintenance by `dbo` only
No `dbo GRANTS` to `<login>`, `<user>`
4 Humans must use real `<login>`;
• may be privileged
Batch jobs must use non-human `<login>`
• unprivileged

Relational Database • OLTP

- Complies with all aspects of Dr E F Codd's *Relational Model* (not the pretenders, not the "theoreticians")
  - Not only with Codd's famous *Twelve Rules*, but all 57 rules in the *Relational Model*
- The database is completely **Normalised**
  - Full **3NF** per Codd, far more than "5NF"
  - Genuine **DKNF** (per Codd intention, not the deranged mathematical definition) is supplied, but this is not commonly understood
  - No Nulls or ambiguities are stored ("6NF" is used in a fuller sense)
- Complete **Data Independence**:
  - The database is truly independent of all applications
  - All database **Consistency**, including **Data Integrity** (both Referential and **Relational Integrity**), is controlled within the database, via Relational Keys & Constraints per Codd intention. All such definitions are **Declarative**
- All updates to the database via ACID **Transactions** only, which can be executed from any application
  - Transactions are implemented as stored procs. This set of stored procs constitute the **Database API** (methods)
  - High Concurrency (low contention) is supplied through **OLTP Standards**
- **Views** are not *required*, however, to provide simplified access to data, a full set of Views is supplied
  - The result is that most readers will access the data via Views, rather than accessing the 'raw' tables
  - Additional Views may be created
- Thus the database is defined Declaratively, and it is completely self-contained, with a view to backup, DR, transport, etc.
- An App Server is recommended for high-volume web applications.
  - For 4GL applications, the **N**on **V**isual **O**bjects are deployed in the application (2 Tier) or Web server (3 Tier)

Relational Database • OLAP

- OLTP and OLAP are supplied from the one database (a separate server; database; and the attendant synchronisation problems; etc, are not required)
  - All nominated Dimensions are supported (if the database is correctly Normalised, the Logical Structure reflects the Dimensions)
  - Pivoting is provided for nominated tables
- Any Report tool may be used directly with the database
  - A high-end Report tool is not required
- Tempdb is not used in the delivered application & reports (it may be used in other applications or reports)

Non-architecture

- The absence of *any* of the above facilities results in a non-compliant implementation, that will be difficult-to-impossible to enhance and expand, depending on the level of non-compliance.
- However, in the common case, the architecture, the deployment of functional components, is entirely absent. Such implementations are **Non-architecture**, they cannot be construed as a "closed architecture" because there is none. Commonly, the app is a single massive software stack, a monolith, with a few rules and constraints, but these are outside the database. The result being a "database" that is accessible only through the app, the opposite of Open Architecture.
- Further, the "database" is in fact a pre-Relational, pre-Hierarchical, 1960's Record Filing System, with **no Relational Integrity, Power or Speed**. This is due to the post-Codd authors, the "theoreticians", who expound such primitive methods, and fraudulently market them as "relational".