

Workflow Tools

Based on: Hollingsworth, The Workflow
Reference Model.

Workflow Management Coalition Document Number TC00-1003 19-Jan-95
<http://www.wfmc.org/standards/docs/tc003v11.pdf>

Definitions (from the *Workflow Management Coalition*)

Workflow:

- “The computerised facilitation or automation of a business process, in whole or part.”

Workflow technology is often an appropriate solution to BPR activities.

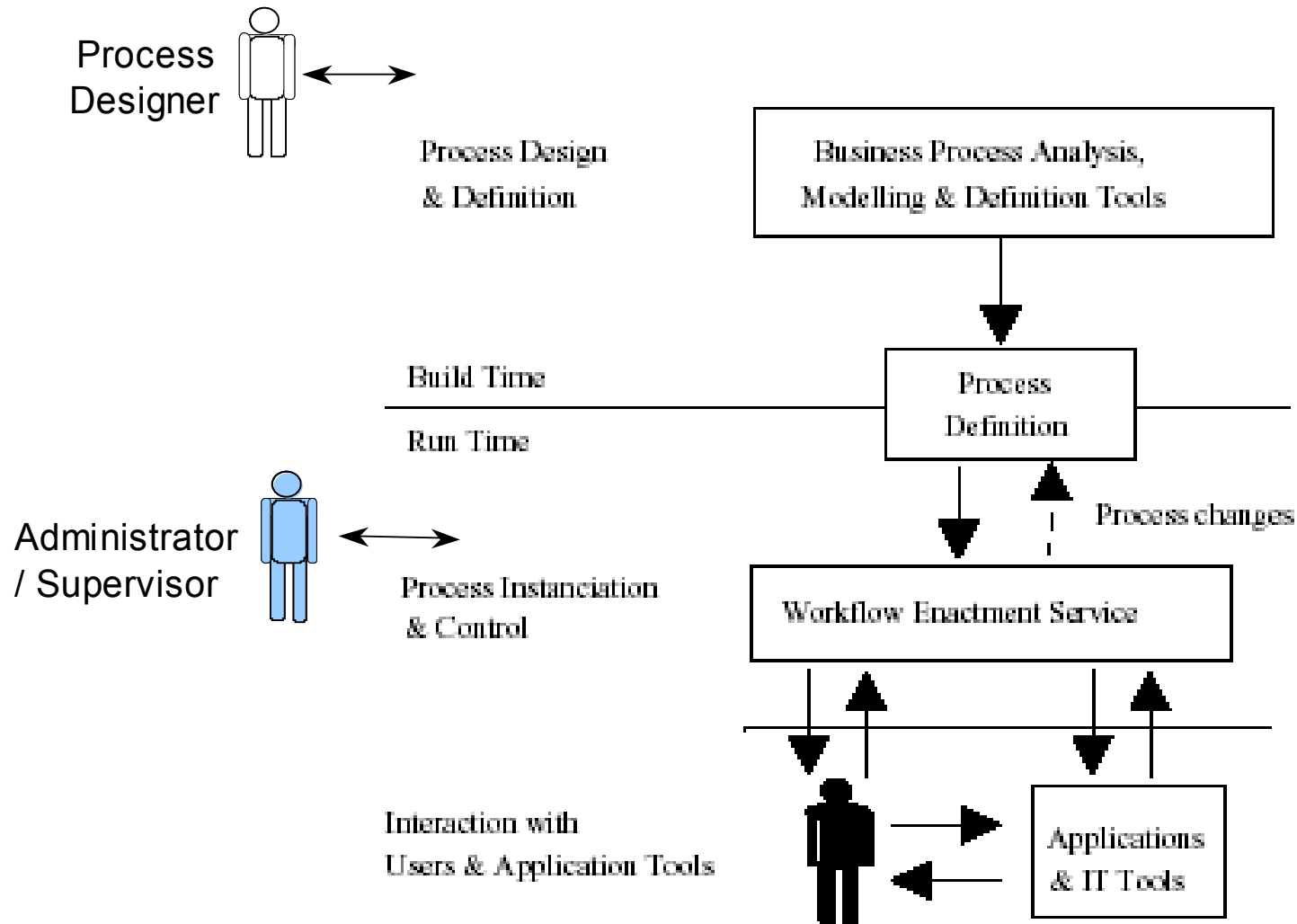
Workflow Management System

- A system that completely defines, manages and executes “workflows” through the execution of software whose order of execution is driven by a computer representation of the workflow logic.

Workflow Management Systems provide support in three functional areas.

- **the Build-time functions**, concerned with defining, and possibly modelling, the WF process and its constituent activities
- **the Run-time control functions** concerned with managing WF processes in an operational environment and sequencing the various activities to be handled as part of each process
- **the Run-time interactions** with human users and IT application tools for processing the various activity steps

Workflow System Characteristics



Build-time Functions

- Result in a computerised definition of a business process.
- Business process is translated from the real world into a formal, computer processable definition by the use of one or more analysis, modelling and system definition techniques.
- Resulting definition is sometimes called a **process model**, a **process template**, **process metadata**, or a **process definition**.

Process Definition

A number of discrete activity steps.

Associated computer and/or human operations.

Rules governing the progression of the process through the various activity steps.

May be expressed in textual or graphical form or in a formal language notation.

Some workflow systems allow dynamic alterations to process definitions at run-time.

Run-time Process Control Functions (1)

Process definition is interpreted by software which is responsible for:

- creating and controlling operational instances of the process
- scheduling the various activities steps within the process
- invoking the appropriate human and IT application resources, etc.

Run-time process control functions link the process as modelled and the process as seen in the real world.

Run-time Process Control Functions (2)

Core component is the basic workflow management control software (or "engine"), responsible for:

- process creation & deletion
- control of the activity scheduling within an operational process
- interaction with application tools or human resources.

Software is often distributed across a number of computer platforms to cope with processes which operate over a wide geographic basis.

Run-time Activity Interactions

Individual activities within a workflow process are typically concerned with:

- human operations, often realised in conjunction with the use of a particular IT tool (e.g. form filling), or
- information processing operations requiring a particular application program to operate on some defined information (e.g. updating an orders database with a new record).

Must interact with the process control software to

- transfer control between activities
- ascertain the operational status of processes
- invoke application tools and pass the appropriate data, etc.

Benefits of a standardised supporting framework include:

- use of a consistent interface to multiple workflow systems
- the ability to develop common application tools to work with different workflow products.

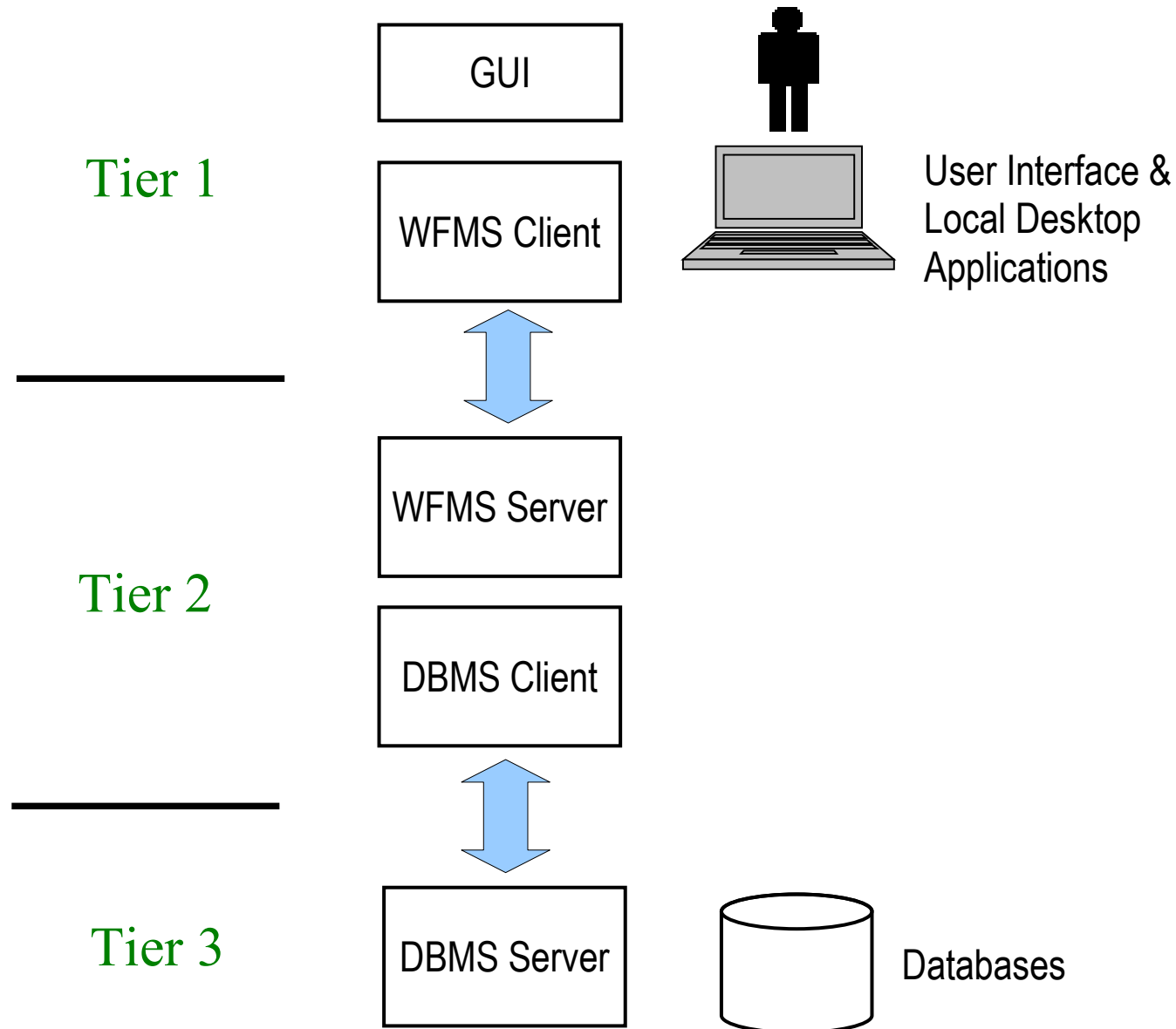
Distribution and System Interfaces

Ability to distribute tasks is a major distinguishing feature of runtime infrastructure.

May operate at a variety of levels

May use a variety of communications mechanisms

WFMS 3 Tier System Structure



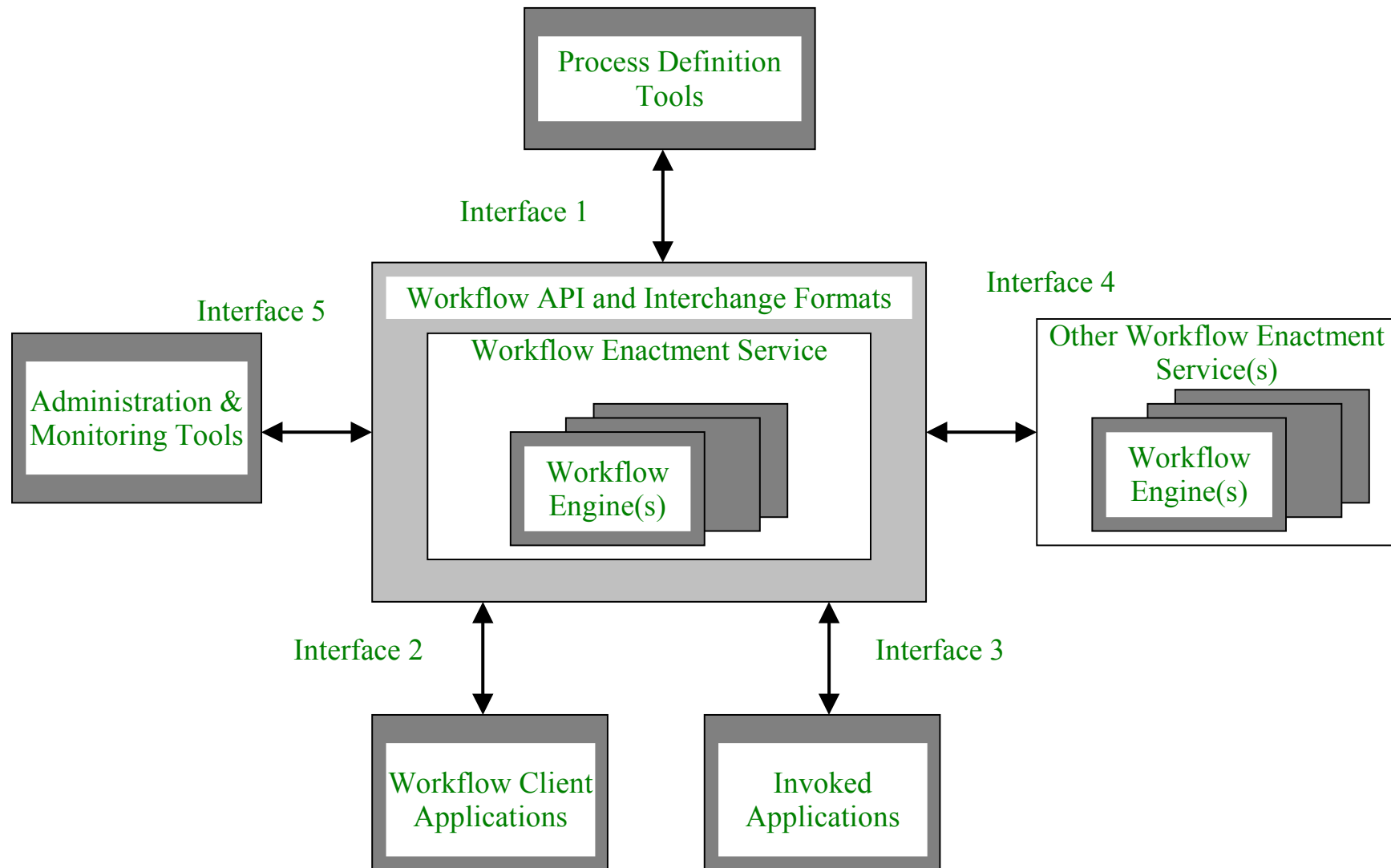
Tiers in Detail

- **User Interface Tier**
 - Deals with all user interactions & focuses their efficient accessibility
 - Usually a GUI on User's desktop
 - Invokes methods on Business Logic Tier & thus acts as client for it
- **Business Logic Tier**
 - Server-based code (made up of business objects that perform logical business operations) with which client interacts
 - Resides at a shared server to optimize allocation of system resources
- **Data Storage Tier**
 - Part of the dist'd application that manages access to data and storage mechanisms eg DBMS.
 - Made up of objects that encapsulate DB routines interacting with DBMS routines directly.

WFMS-DBMS: Compare & Contrast

Component	DBMS	WFMS
<u>Meta-Model:</u> Constructs & Associated functions	(Relational DBMS) Table with rows & columns & SQL operations	No Industry-Wide Standard
<u>Build-Time:</u> Functions to define user specified constructs in terms of the meta model	Functions provided by DDL	Functions to define process models, organizational structures, applications, servers & networks
<u>Run-Time:</u> Operations on user specified constructs	Functions provided by DML	Creating, navigating & controlling processes
<u>Database:</u> Store of info managed by build & runtime components	System Catalog & User defined tables	User-defined constructs (e.g. process models) and instances of these constructs (instances)

Workflow Reference Model - Components & Interfaces



WFMS User Types

- End Users
- Process Modellers (Business Analysts)
- Process Administrators
- System Administrators
- Customer Support

Interoperability

Flow of work may involve transfer of tasks between different vendors' workflow products so that different parts of the business process can be enacted on different platforms.

Flow within the central box passes between two or more workflow products - for example activities 1,2 and 5 may be executed by one workflow system and 3 and 4 by a different system, with control passed between them at appropriate points within the overall workflow.

Standards to support this transfer of workflow control enable the development of composite workflow applications using several different workflow products operating together as a single logical entity.

Range of interface definitions

- specifications for process definition data and its interchange
- interfaces to support interoperability between different workflow systems
- interfaces to support interaction with a variety of IT application types
- interfaces to support interaction with user interface desktop functions
- interfaces to provide system monitoring and metric functions to facilitate the management of composite workflow application environments

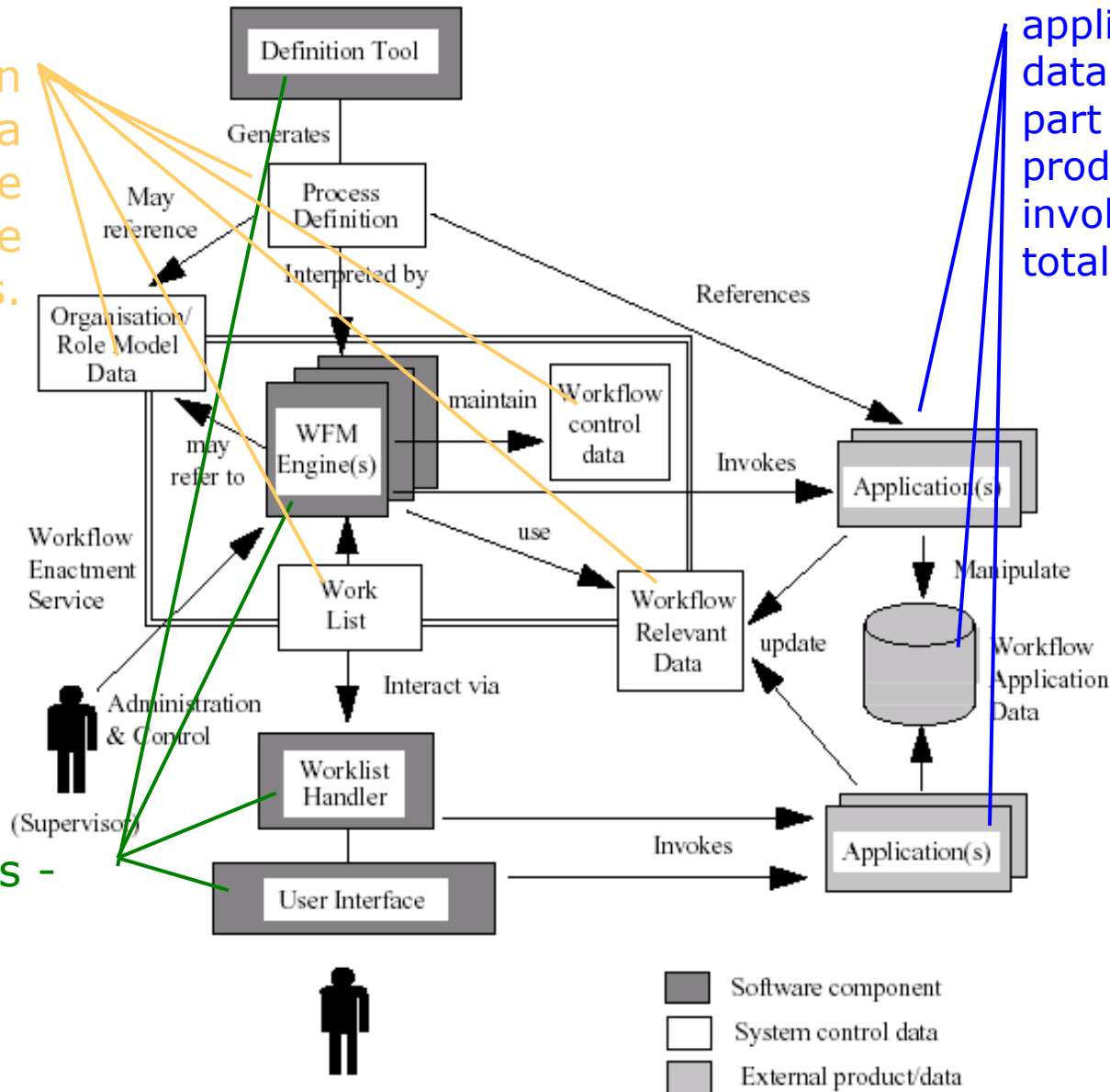
Product Implementation Model

- Despite the variety in WF products, possible to construct a general implementation model of a WF system which can be matched to most products in the marketplace.
- This approach identifies main functional components within a WF system and the interfaces between them as an abstract model.
- Many different concrete implementation variants of this abstract model will exist and therefore the interfaces specified may be realised across a number of different platform and underlying distribution technologies.
- Not all vendors want to expose every interface between the functional components within the model; this is dealt with by having a variety of conformance levels which will identify the particular interworking.

Generic Workflow Product Structure

system definition and control data used by one or more software components.

applications and application databases which are not part of the workflow product, but which may be invoked by it as part of the total workflow system

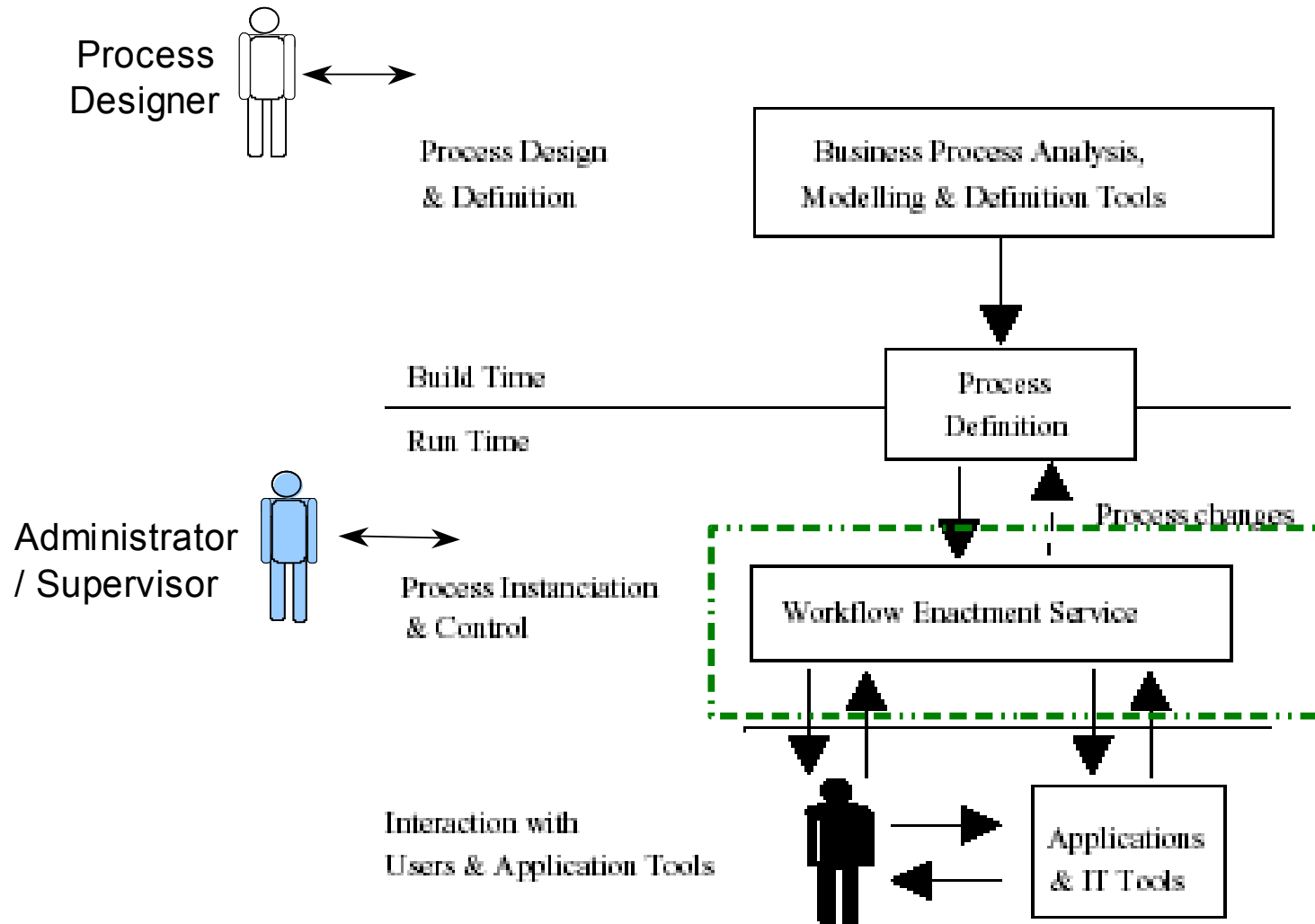


software components - provide support for functions within the workflow system.

Workflow Reference model

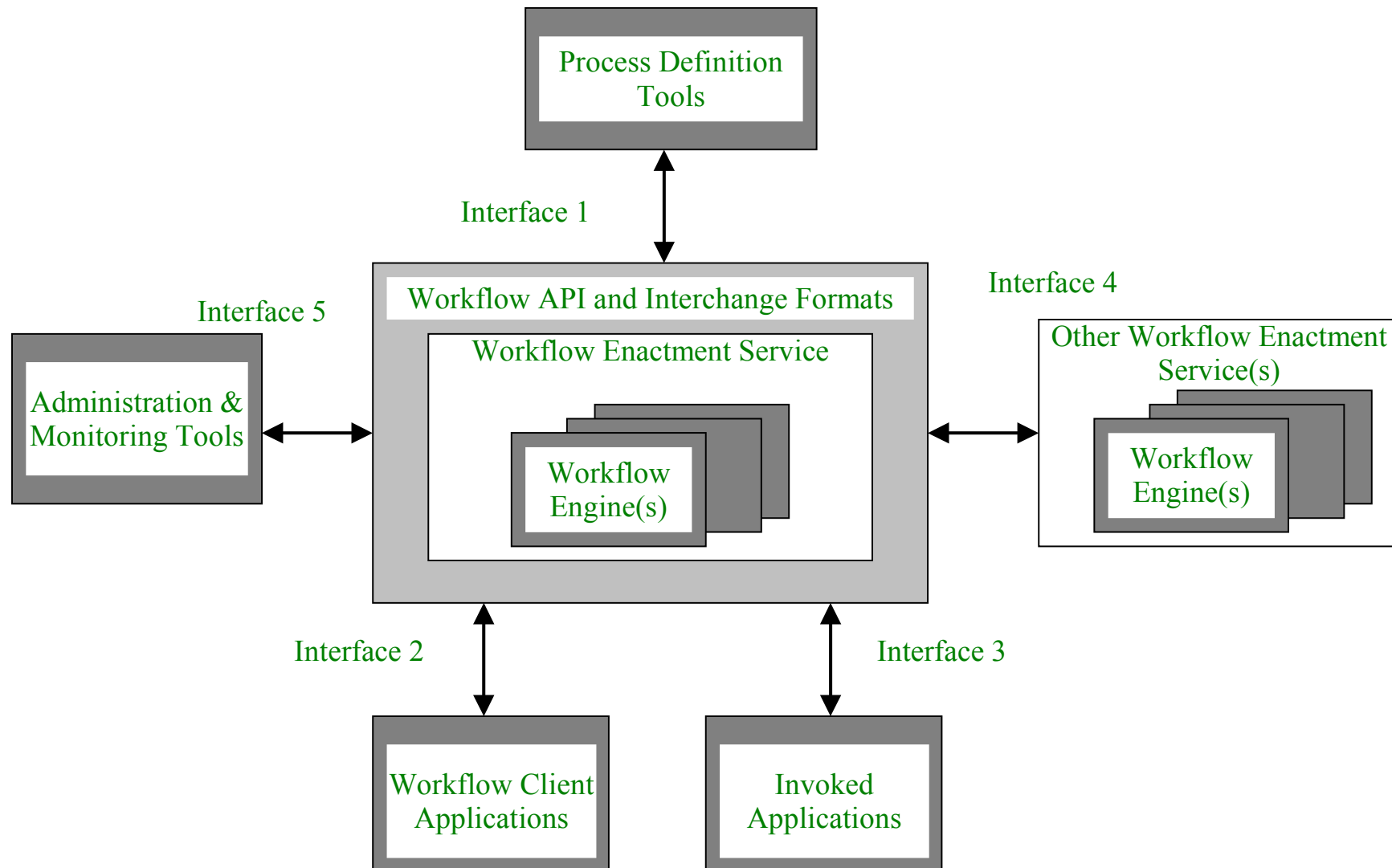
- Developed from the generic WF application structure by identifying the interfaces that allow products to interoperate at a variety of levels.
- All WF systems contain generic components which interact in a defined set of ways; different products exhibit different levels of capability within each of these generic components.
- Need a standardised set of interfaces and data interchange formats between components.
- Distinct interoperability scenarios can be constructed by reference to the interfaces, - different levels of functional conformance as appropriate.

Workflow System Characteristics



WF Ref Model:
All Interfaces
Interacting with
WF Enactment
Service

Workflow Reference Model - Components & Interfaces



Workflow Enactment Service

A software service that may consist of **one or more workflow engines** in order to create, manage and execute workflow instances.

Applications may interface to this service via the workflow application programming interface (WAPI).

Workflow Engine Functions

A software service or "engine" that provides the run time execution environment for a workflow instance.

Typical facilities:

- interpretation of the process definition
- control of process instances - creation, activation, suspension, termination, etc
- navigation between process activities, which may involve sequential or parallel operations, deadline scheduling, interpretation of workflow relevant data, etc
- sign-on and sign-off of specific participants
- identification of workitems for user attention and an interface to support user interactions
- maintenance of WF control data and WF relevant data, passing WF relevant data to/from applications or users
- an interface to invoke external applications and link any WF relevant data
- supervisory actions for control, administration and audit purposes

Homogeneous V Hetero-geneous WF Enactment Services

•Homogeneous:

- 1/ more compatible WF engines (ie support same metamodel) providing the runtime execution envt for WF processes with defined set of process definition attributes.
- Protocols and interchange formats used are product specific and not standardised.

•Heterogeneous:

- 2/more homogeneous services, following common standards (ie common denominator needed in terms of interfaces) for interoperability at a defined conformance level
 - one of:
 - + A common naming scheme across the heterogeneous domain
 - + Support for common process definition objects & attributes across the domain
 - + Support for WF relevant data transfer across the domain
 - + Support for process, sub-process or activity transfer btw heterogeneous WF engines
 - + Support for common admin & monitoring functions within the domain

Process and Activity State Transitions

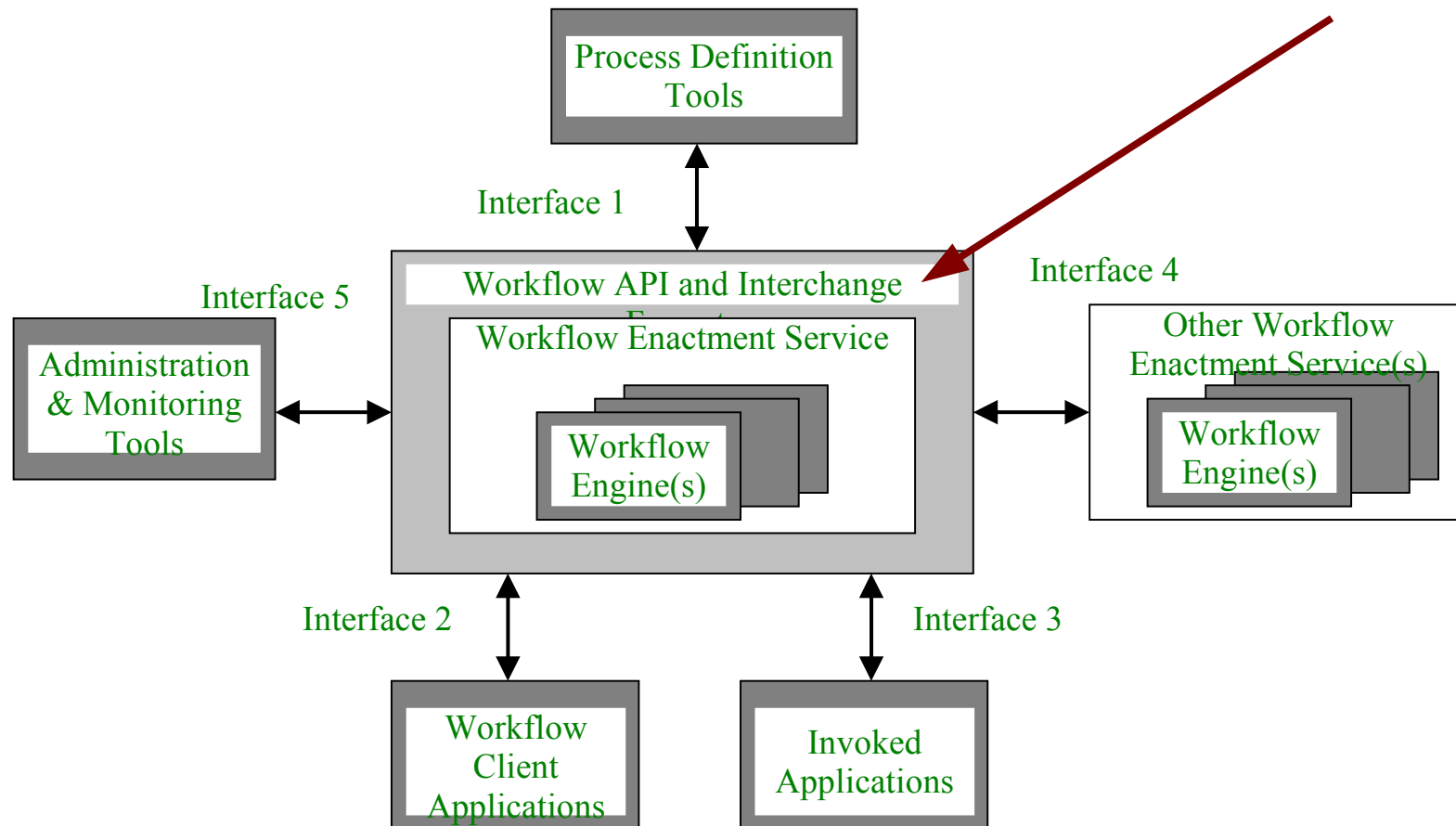
The workflow enactment service may be considered as a state transition machine.

Individual process or activity instances change states in response to:

- external events (e.g. completion of an activity) or
- specific control decisions taken by a workflow engine (e.g. navigation to the next activity step within a process).

Workflow Application Programming Interface & Interchange (WAPI)

Set of API calls and interchange functions supported by a WF enactment service at its boundary for interaction with other resources and applications.



Workflow Control Data

Definition:

Internal data that are managed by the workflow management system and/or workflow engine.

The workflow enactment service maintains internal control data to identify the state of individual process or activity instances and may support other internal status information. These data are not accessible or interchangeable, but some of the information content may be provided in response to specific commands.

Workflow Relevant Data

Definition

Data that are used by a workflow management system to determine the state transition of a workflow process instance.

- May affect the choice of the next activity to be executed.
- Such data are potentially accessible to WF applications for operations on the data and thus may need to be transferred between activities by the WF enactment software.
- When operating in a heterogeneous environment, such data may need to be transferred between WF engines, where the process execution sequence spans two or more WF engines.

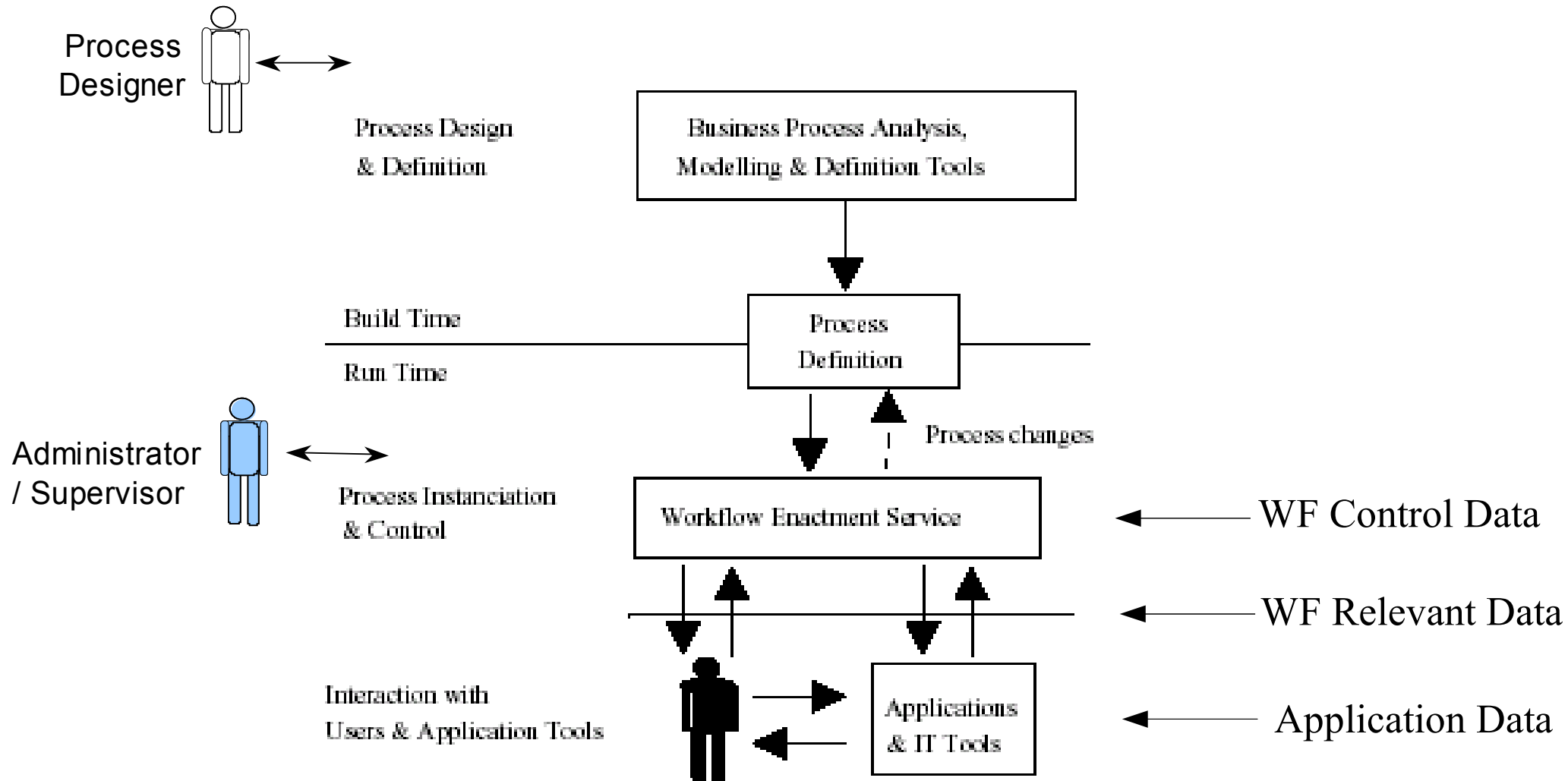
Workflow Application Data

Definition

Data that are application specific and not accessible by the workflow management system.

- Manipulation of application data may be required within each activity of a process.
- WF model must cope with interchange of case data between the various activities.
- May also require transformation of case data between different tool data formats, for example conversion of a document or spreadsheet from one application format to another.
- WF application data are not used by the workflow enactment software.

Workflow System Characteristics



Interface 1: Process Definition

Process Definition Tools

- Variety of tools available. The WF model is not concerned with the particular nature of such tools nor how they interact.
- Process definitions defined within a WF product will normally be held within the WF product domain and may, or may not, be accessible.
- Where separate products are used for defining the process, the process definitions may be transferred between the products as and when required or may be stored in a separate repository, accessible to both products (and possibly other development tools).

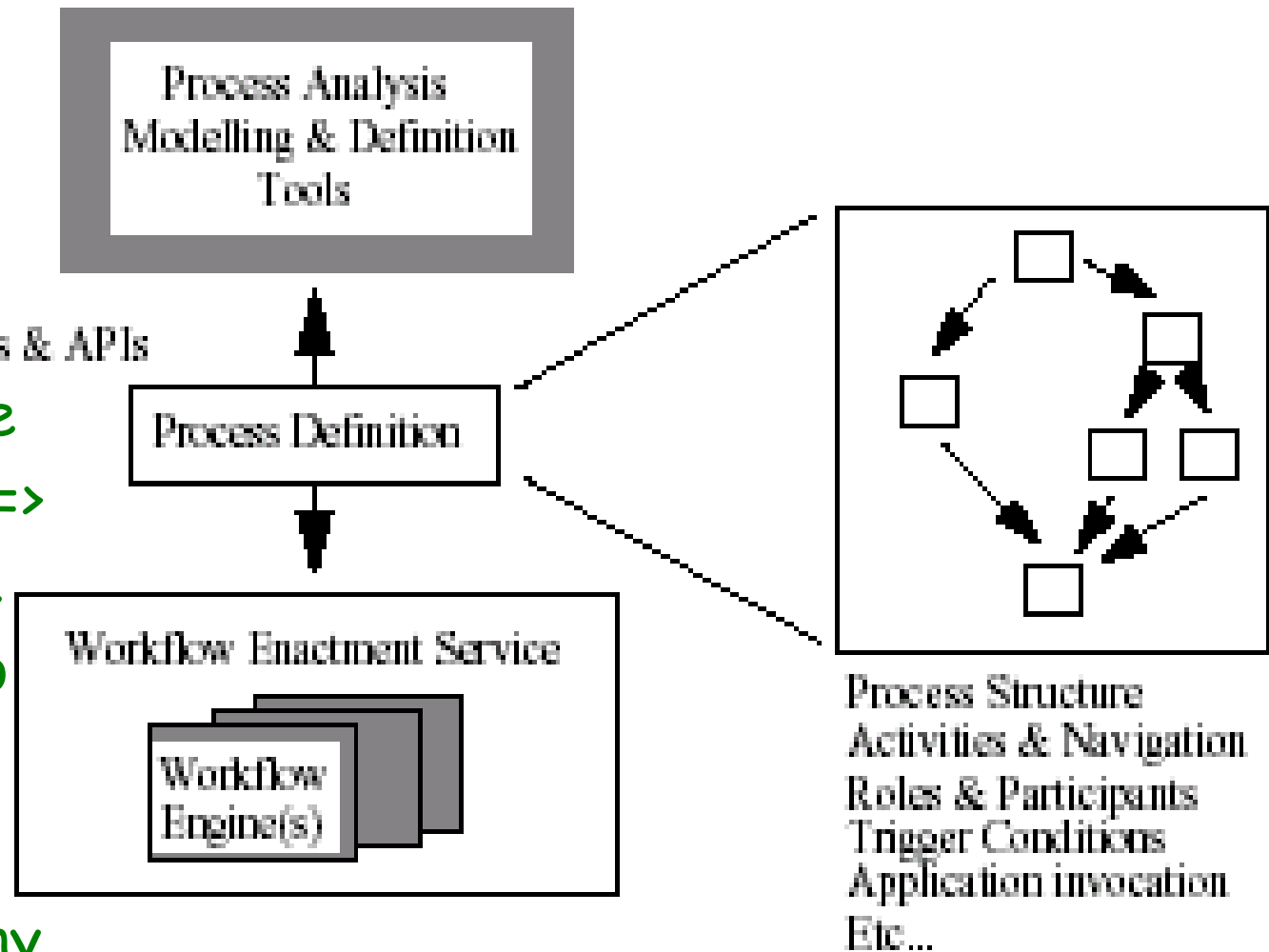
Process Definition Tools

- Final output from process modelling and design is proc def which can be interpreted at runtime by the WF engine(s) within the enactment service. Each individual process definition is typically in a form specialised to the particular WFMS for which it was designed. The WF definition interchange interface will enable more flexibility in this area.
- Process analysis, modelling and definition tools may include the ability to model processes in the context of an organisation structure (although this is not a mandatory aspect of the WF reference model). Where an organisation model is incorporated into such tools the process definition will include organisation related objects such as roles. These are related (typically) to system control data such as role: actor relationships (eg within an organisational directory) which may be referenced during process execution.

Workflow Definition Interchange (Interface 1)

- Advantages of standardised form of interface:

- Separation btw buildtime & runtime environments => allows proc def from one modelling tool as input to other runtime products
- Offers the potential to export a proc def to many WF products to co-operate & provide a distributed service



Interface 4: Interoperability: Java WFMS Implemented in RMI

- Use Remote Message Invocation (RMI) to support WFMS
 - JWFMS uses RMI (more below) as a communication protocol impl'ing the 3 Tier Architecture between Client and Server.
 - RMI takes advantage of Java's platform independence for simplicity of design and reduction of maintenance demand.
 - Other protocols such as Common Object Request Broker Architecture (CORBA), MS DCOM or Enterprise Java Beans (EJB) offer more functionality at the expense of added complexity.
- Remote Message Invocation (=RPC in Java)
 - Supported by two java packages: `java.rmi` & `java.rmi.server`.
 - An application that uses RMI has 3 components:
 - + An interface that declares headers for remote methods;
 - + A server class that implements the interface; and
 - + One or more clients that call the remote methods.

JWFMS Users

- JWFMS supports 3 kinds of users:
 - System Admin (akin to UNIX Root User):
 - + with all access privileges to monitor workflows & setup system parameters
 - + Can add/delete/modify user info
 - + Can create/delete/modify workflow info
 - Workflow Manager
 - + Limited privileges Admin using JWFMS mgmt client
 - + Can view content & modify structure of WF that (s)he is manages
 - + Set transactions' start time & monitor, retrieve & modify data assoc with WF
 - Users:
 - + Perform transactions in the WF
 - + Access the system thro the JWFMS user client
 - + Have rights to view/manage own info but not other users' info

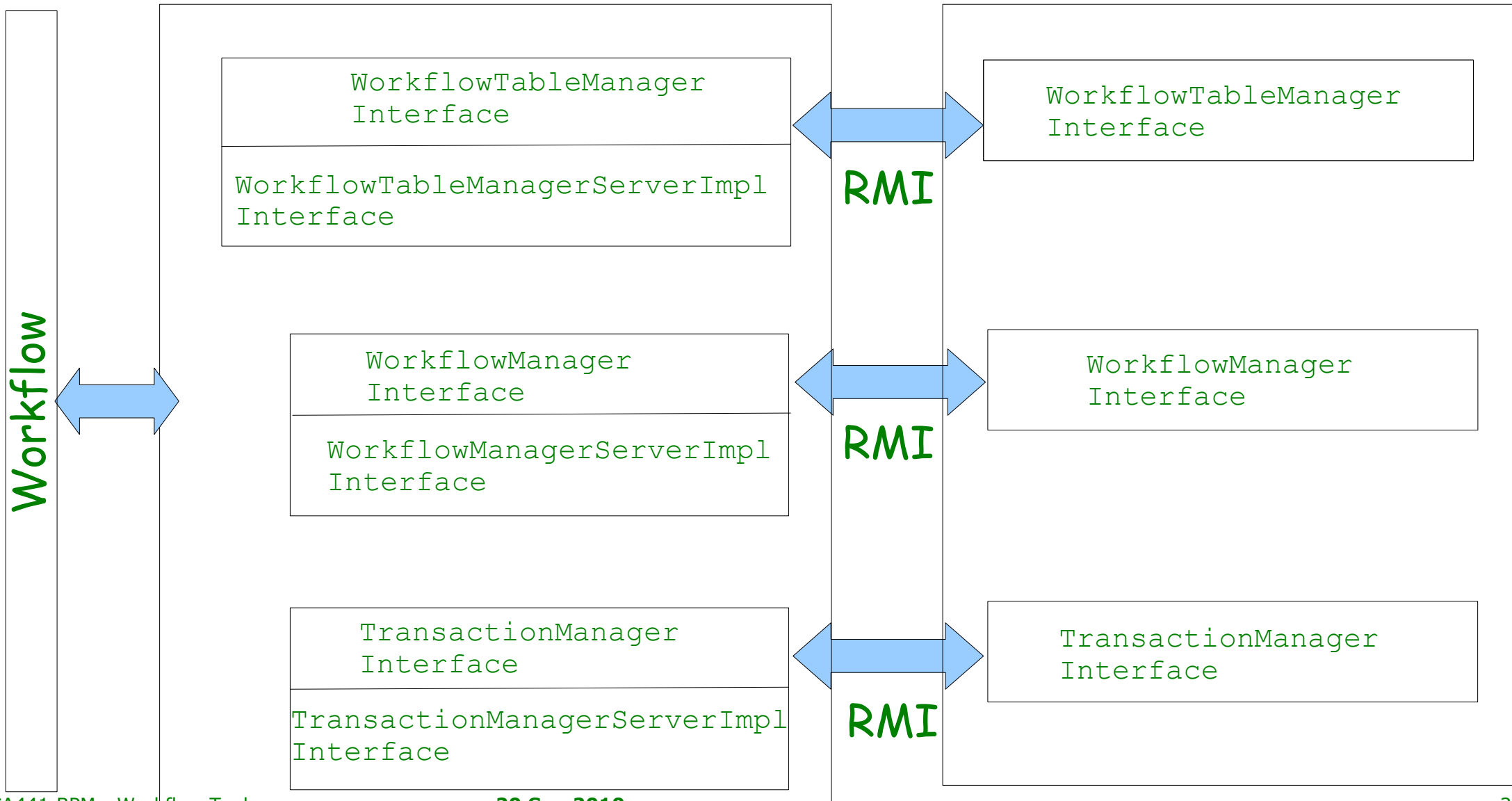
Components of JWFMS

- Clients/Server communicate thro dist'd RMI interface classes
- These classes function like C/S APIs & called `XYZManager` with methods allowing clients access to system objects (eg Wfs,transactions, users etc)
- Client
 - + first acquires client instance of relevant manager class thro RMI protocol
 - + invokes public method to retrieve java object for info client needs to operate
 - + e.g. If client wants to modify WF uses `WorkflowManager` interface class to retrieve `Workflow` java object
 - + When changes need to be saved, client uses `updateWorkflow()` method in `WorkflowManager` interface class
- Manager Interface classes needed for Server API for WF Mgmt:
 - + `WorkflowTableManager` Interface (stores info on WF activities in a table)
 - + `WorkflowManager` Interface (executes/automates WF activities)
 - + `TransactionManager` Interface (guarantees atomicity of transactions)
 - + Thus each of these export functions thro the various Interface classes

Components of JWFMMS: Graphical

Server Side

Client Side



References

- David Hollingsworth "The Workflow Reference Model", Workflow Management Coalition Document Number TC00-1003 January-95 <http://www.wfmc.org/standards/docs/tc003v11.pdf> (also in "Workflow handbook 2004" ed. Layna Fischer, Lighthouse Point 2004)
- David Hollingsworth "The Workflow Reference Model 10 Years On", in the "Workflow Handbook 2004", Workflow Management Coalition.
http://www.wfmc.org/standards/docs/Ref_Model_10_years_on_Hollingsworth.pdf

The web pages (above) now require you to register with the Workflow Management Coalition, and login, before you can gain access. Registration is free, at <http://www.wfmc.org/>.

- Wang Shaofeng, "The role of Java RMI in Designing Workflow Management System", ACM Sigsoft, Software Engineering Notes Vol 26 no 2, March 2001, p 49
- Wang Shaofeng & Sun Jiaguang, "A Framework Design of Workflow Management System with Java RMI, ACM SIGPLAN Notices, Vol 36 No 9 Sept 2001