DB2 Viper 2: Workload Management

Paul Bird (pbird@ca.ibm.com)
Senior Technical Staff Member
DB2 for Linux, Unix, and Windows

ON DEMAND BUSINESS™

Objectives

- Quickly review the background and strategy for the new capabilities being introduced
- Provide a basic understanding of each of the new capabilities and how they work together
- Provide examples of how these new capabilities can be used to help solve different workload management (WLM) scenarios
- Discuss the primary differences between Query Patroller and DB2 Workload Manager and how they interact in Viper 2

Caveats

- Some simplifications have been introduced and important details left out for the sake of brevity and clarity
  - ***This is an introduction to some of the new capabilities and is not meant to be a comprehensive tutorial***

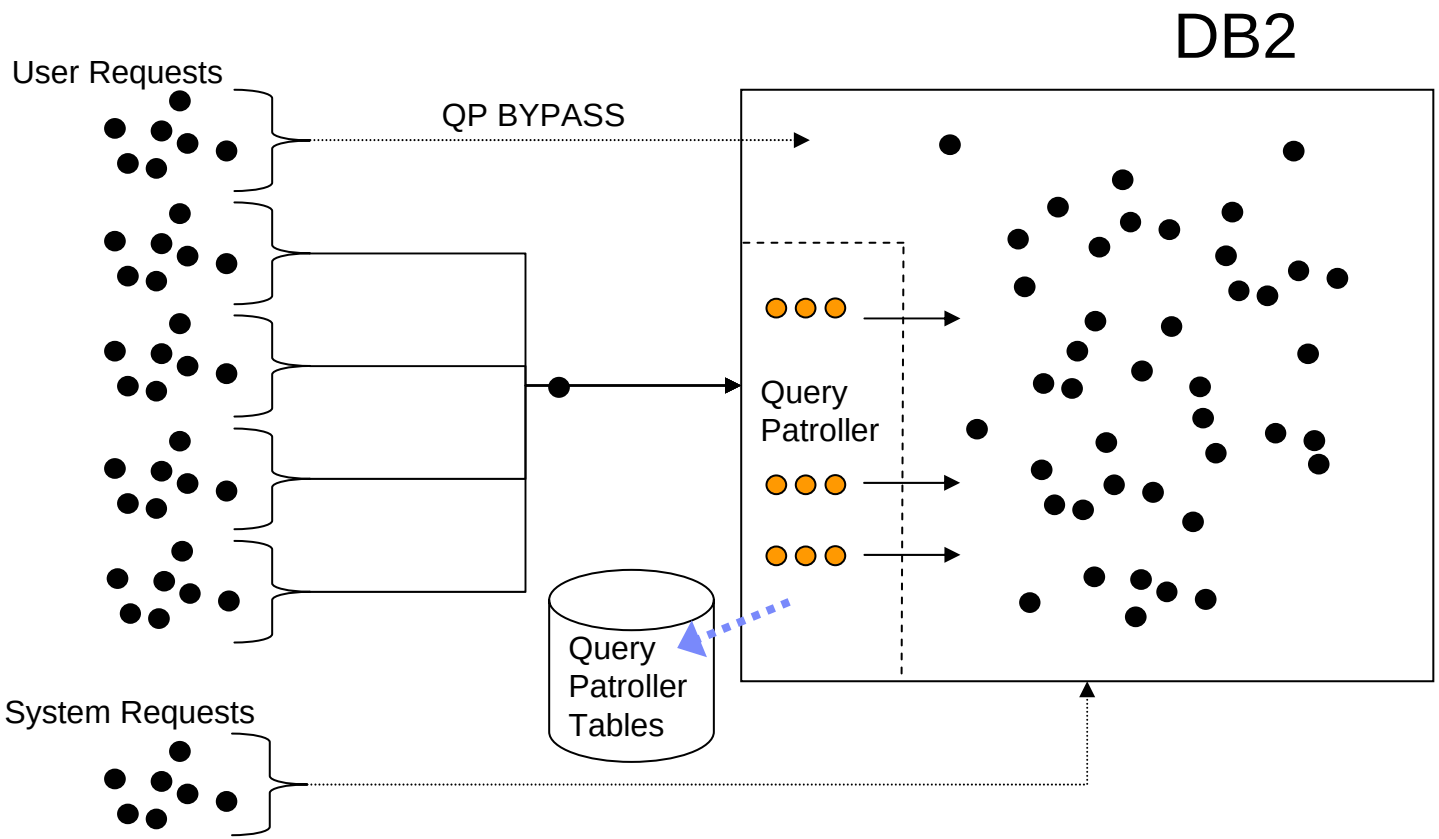
- The focus of this presentation is on introducing the technical aspects of the new engine capabilities for workload management in the Viper 2 release
  - It will ***not*** be covering other parts of the overall WLM solution to be offered by IBM including: packaging, pricing, related tools, documentation, education, etc

Agenda

- Background
- Foundation Concepts
- Monitoring Features: Execution Environment
- Monitoring Features: Activity Information
- Advanced Features
- Query Patroller and Workload Manager

Background

DB2 9.1 Execution Environment (with QP)



DB2
Governor
Monitoring
(ongoing)

Primary Customer Requirements

- A stable, predictable execution environment
- A light-weight, granular way to monitor active work
- Better resource management
  - Be able to explicitly allocate resources amongst work
  - Be able to limit excessive, unexpected resource consumption
- Better request management
  - Be able to manage work based on its business priority
  - Be able to track performance of work
- End-to-end workload management solutions

Strategic Decisions

- Focus on the database execution environment to provide our customers with increased insight and control over active work
- Incorporate base WLM capabilities into the core DB2 engine infrastructure in order to achieve the degree of control and monitoring required with minimal overhead
- Enable tighter integration with other products such as platform workload management offerings

DB2 Viper 2 Objectives for WLM

1. Implement the foundation for new WLM strategy
   - Offer a viable alternative WLM solution to that offered by Query Patroller and DB2 Governor

2. Provide some immediate, tangible relief to our customers in known problem areas
   - Enabling explicit allocation of CPU priority for executing work
   - Controlling “rogue” queries
   - Monitoring of database activity

Foundation Concepts

ON DEMAND BUSINESS™

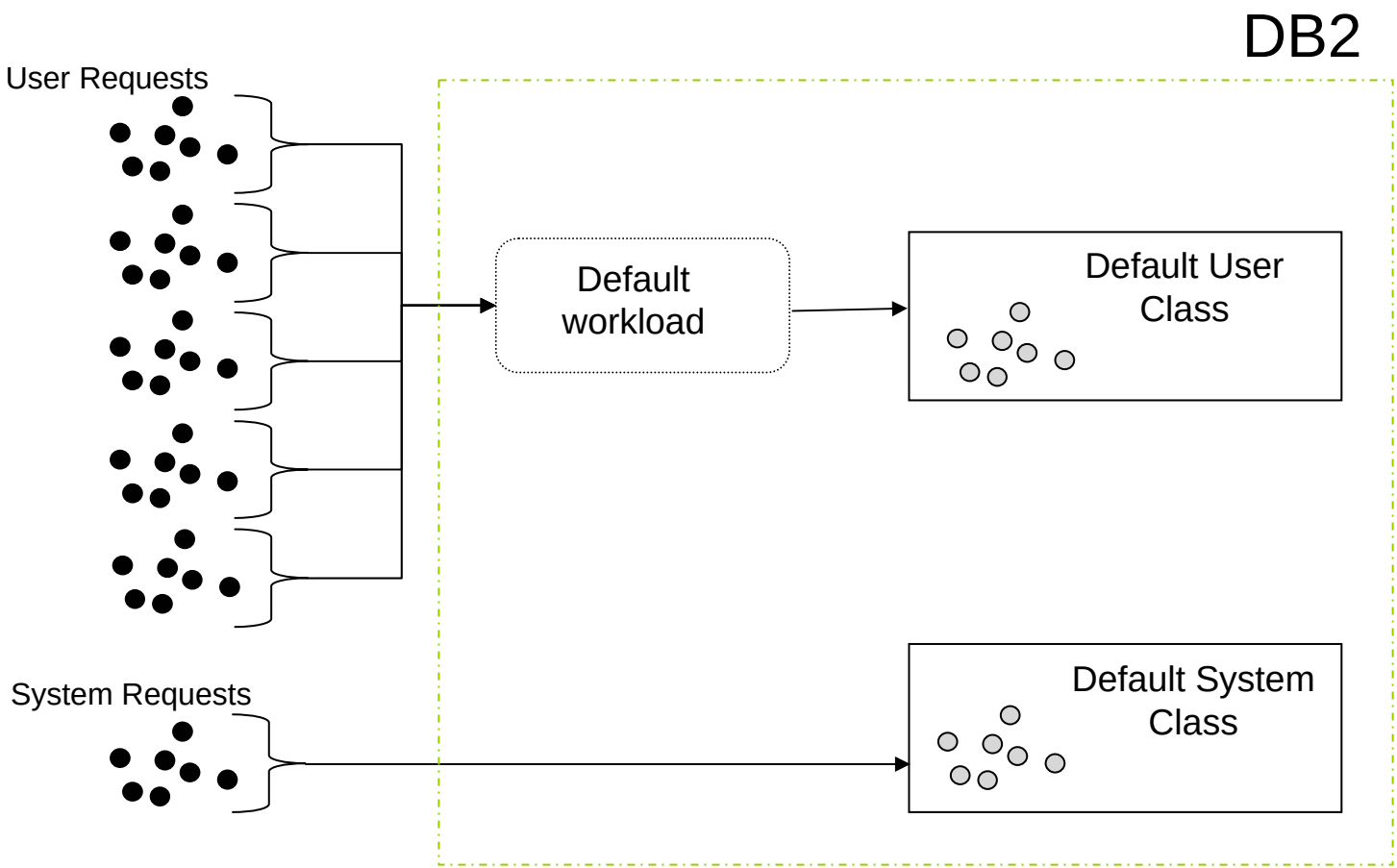
Introducing The DB2 Service Class

- Serves as the primary point of resource assignment, control, and monitoring for work executing within DB2
- All work in a database executes within a DB2 service class
  - Default User Service Class created automatically in Viper 2
- Supports a two-tier hierarchy consisting of super and sub classes
  - A super class is a logical entity providing common attributes across sub classes

Introducing The DB2 Workload

- Serves as the primary point of identity and control for submitters of work to the system
- Directs incoming work to a DB2 service class
- All connections to a database map to a DB2 workload
  - Default Workload created automatically and maps to Default User Service Class
- Connections are mapped to a specific workload when established and re-evaluated at unit of work boundaries as required

Default DB2 Viper 2 Execution Environment



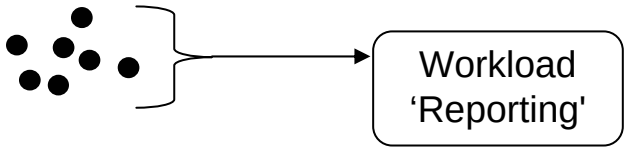
Customization: DB2 Workload

- A DB2 workload can be created to uniquely identify any connections of interest
  - Provides the ability to independently monitor and control them (and their activities)
- New workloads are defined by providing mapping values for a set of connection attributes
  - Application name
  - SYSTEM_USER
  - SESSION_USER
  - Any group of SESSION_USER
  - Any role of SESSION_USER
  - CLIENT USERID
  - CLIENT APPLNAME
  - CLIENT WRKSTNNAME
  - CLIENT ACCTNG
- Can modify the evaluation order of defined workloads

Examples of Defining a DB2 Workload

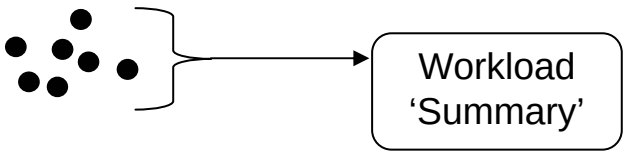
CREATE WORKLOAD “Reporting” APPLNAME(‘Accounts’)

All connections with application name of ‘Accounts’

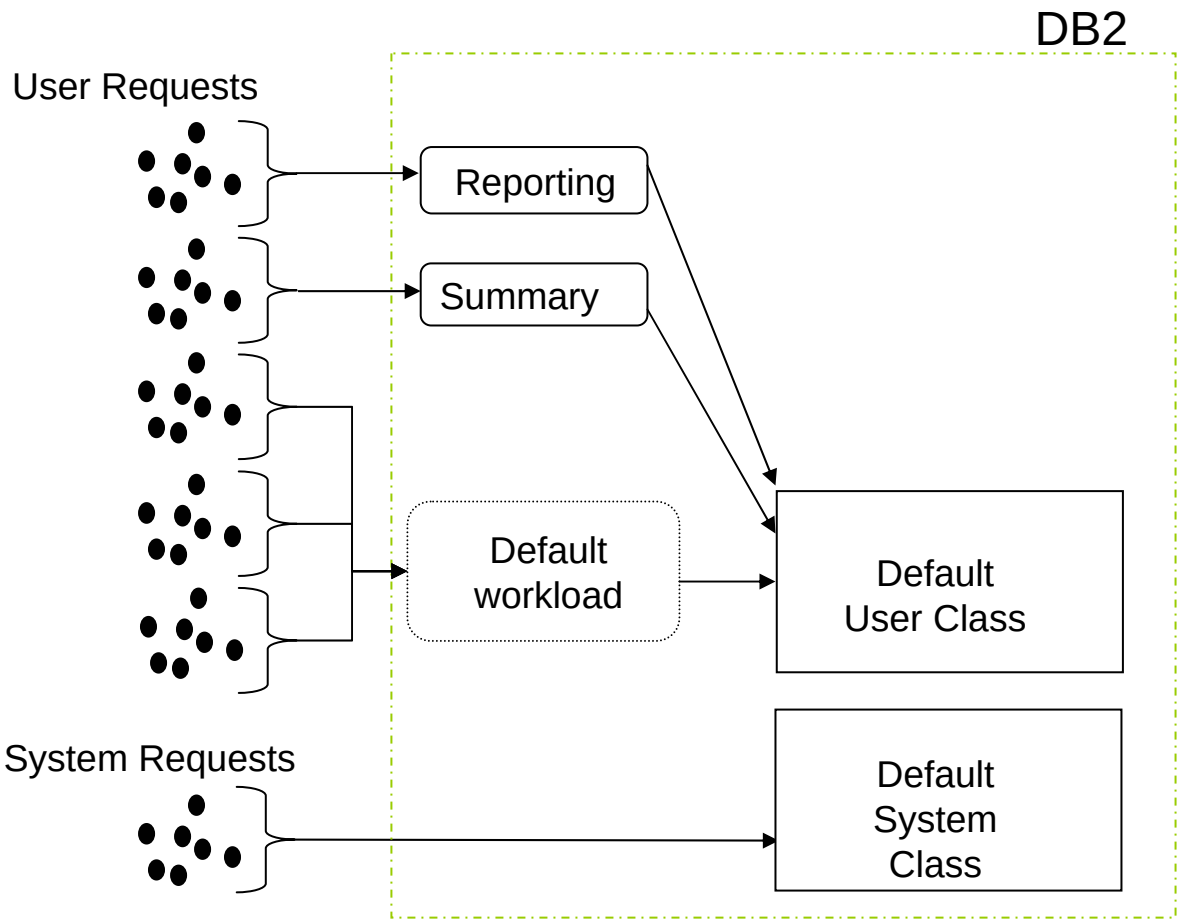


CREATE WORKLOAD ”Summary” SESSION_USER GROUP(‘Deptmgr’)
APPLNAME(‘Accounts’)

All connections with application name of ‘Accounts’ where session user belongs to ‘Deptmgr’ group



Customized Environment: DB2 Workload

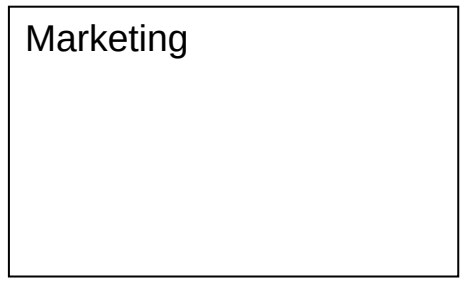


Customization: DB2 Service Class

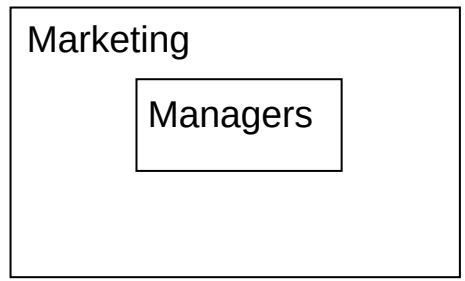
- A DB2 service class can be created to act as a unique execution environment for any grouping of work
  - Provides the ability to independently monitor and control this group of work
  - Can assign different resource priorities to each service class
- Can create a two-tier hierarchy of service classes using super and sub classes
  - Allows for more complex division of execution environment and better emulation of real world model
- Can modify the resources available to a service class
  - CPU Priority
  - Prefetch I/O Priority

Examples of Defining a DB2 Service Class

CREATE SERVICE CLASS ”Marketing”

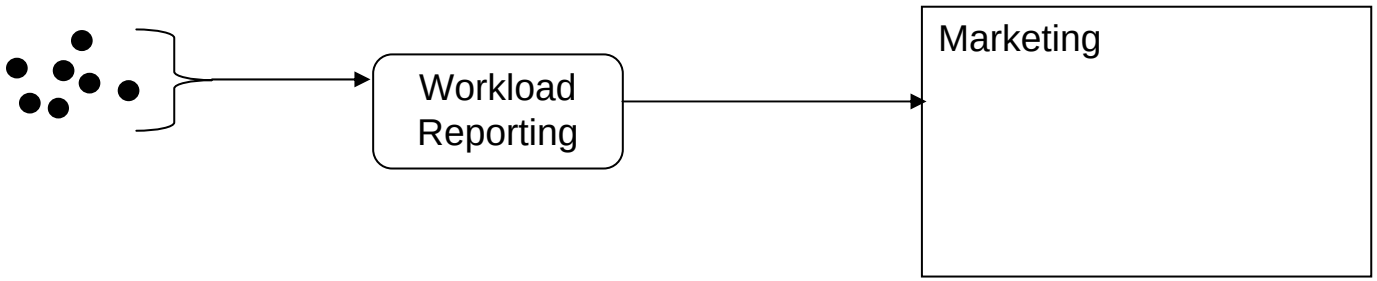


CREATE SERVICE CLASS “Managers”
UNDER “Marketing”

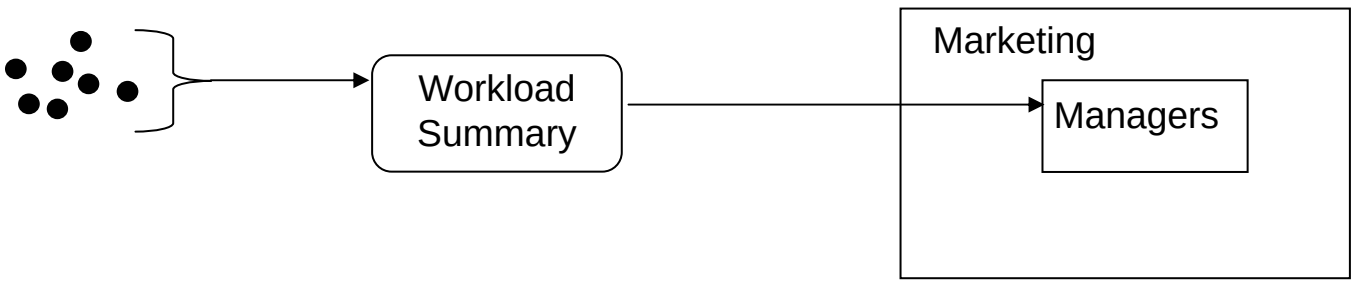


Examples of Mapping To DB2 Service Classes

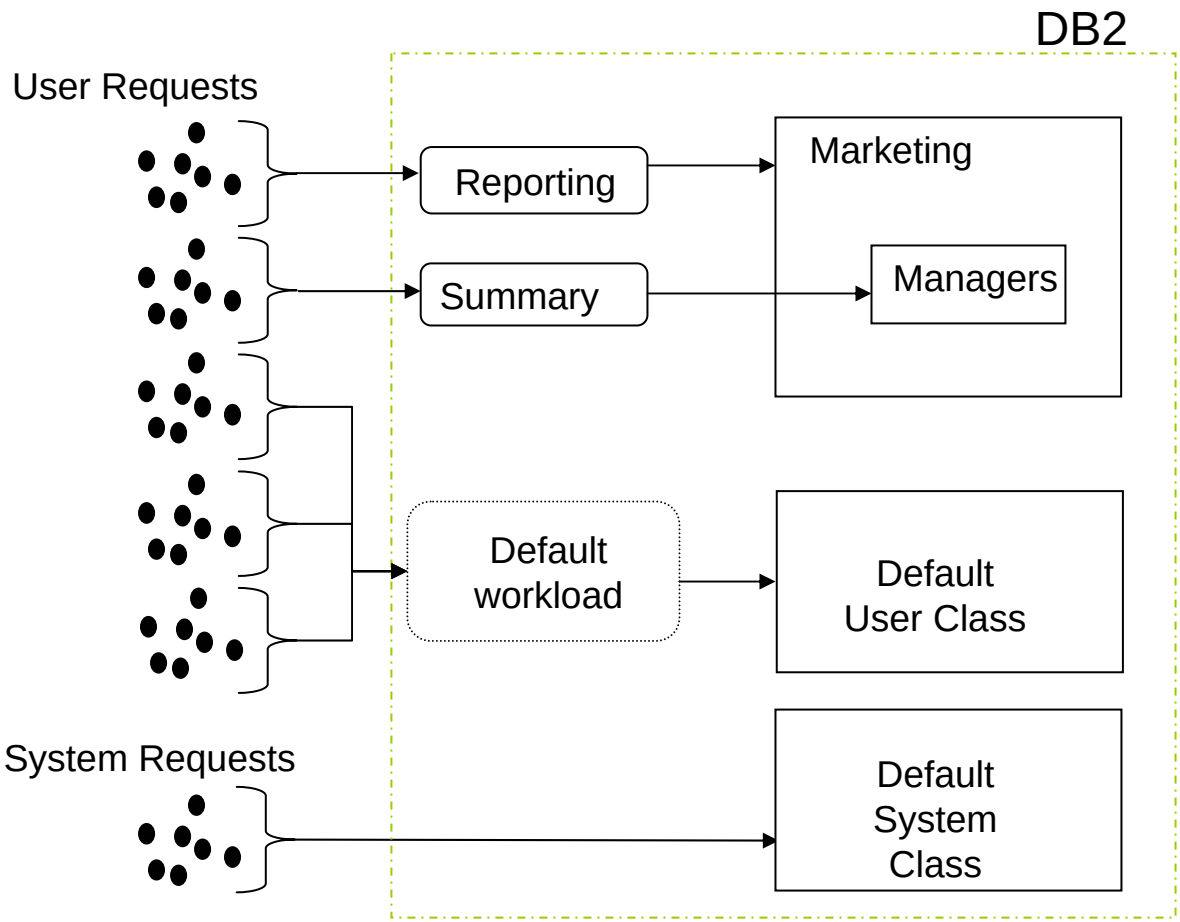
ALTER WORKLOAD “Reporting” SERVICE CLASS “Marketing”



ALTER WORKLOAD “Summary” SERVICE CLASS “Managers”
UNDER “Marketing”



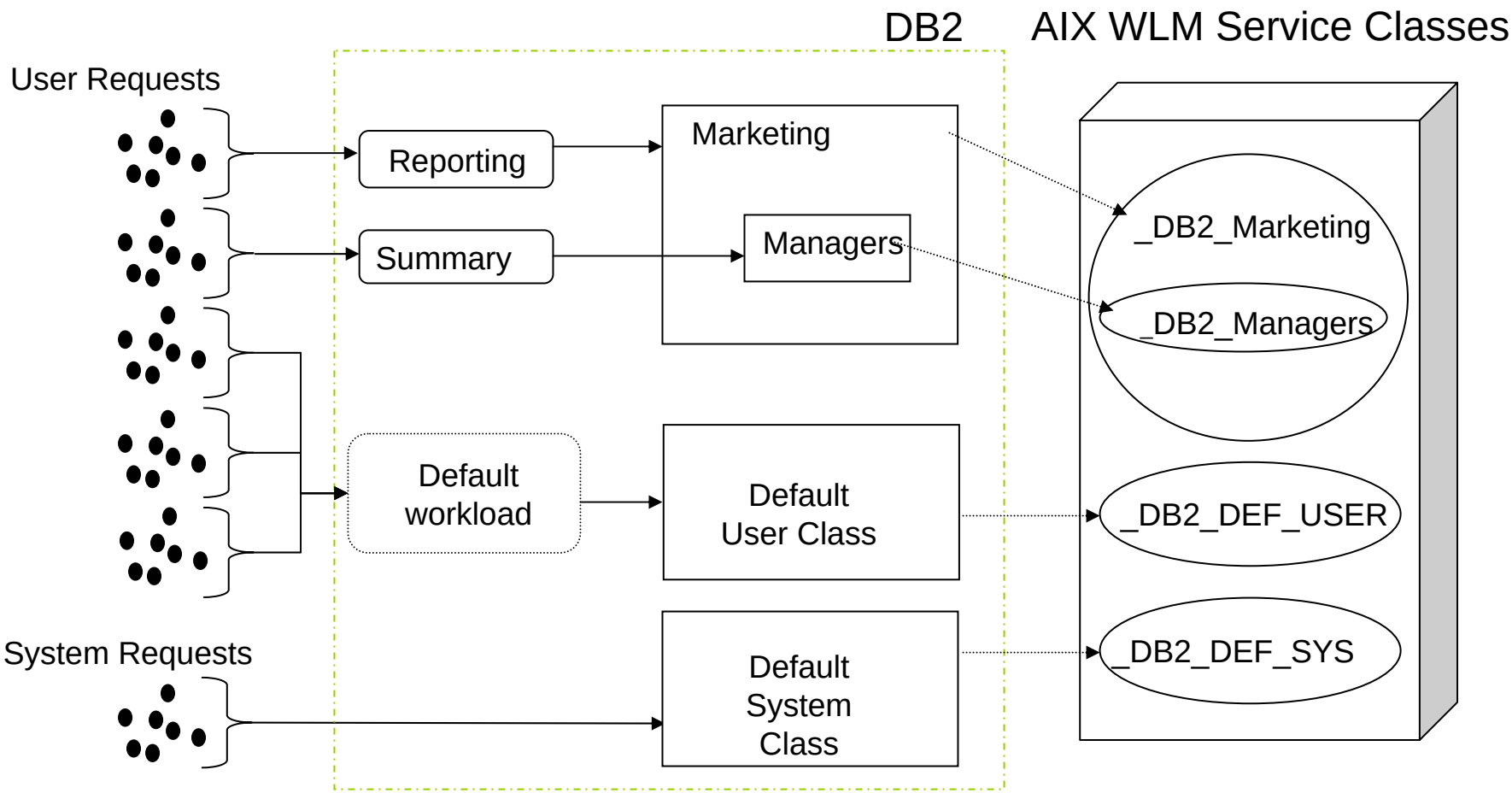
Customized Environment: DB2 Service Class



Integrating DB2 and AIX WLM

- DB2 Service Classes offer the optional ability to tightly integrate DB2 processing with AIX WLM service classes
  - Each DB2 Service Class can be associated with a specific AIX WLM Service Class
    - All agents working in a DB2 service class will automatically be associated by DB2 with the identified AIX WLM Service Class
- AIX WLM provides sophisticated management of CPU to maximize utilization of that resource while respecting any explicit allocations
  - Allocation of CPU done via CPU shares assigned to AIX WLM service classes
  - CPU shares can be adjusted dynamically using AIX WLM interfaces
  - Provides the ability to borrow unused CPU shares from other service classes
  - Provides OS level statistics per AIX service class

Customized Environment: AIX WLM Integration



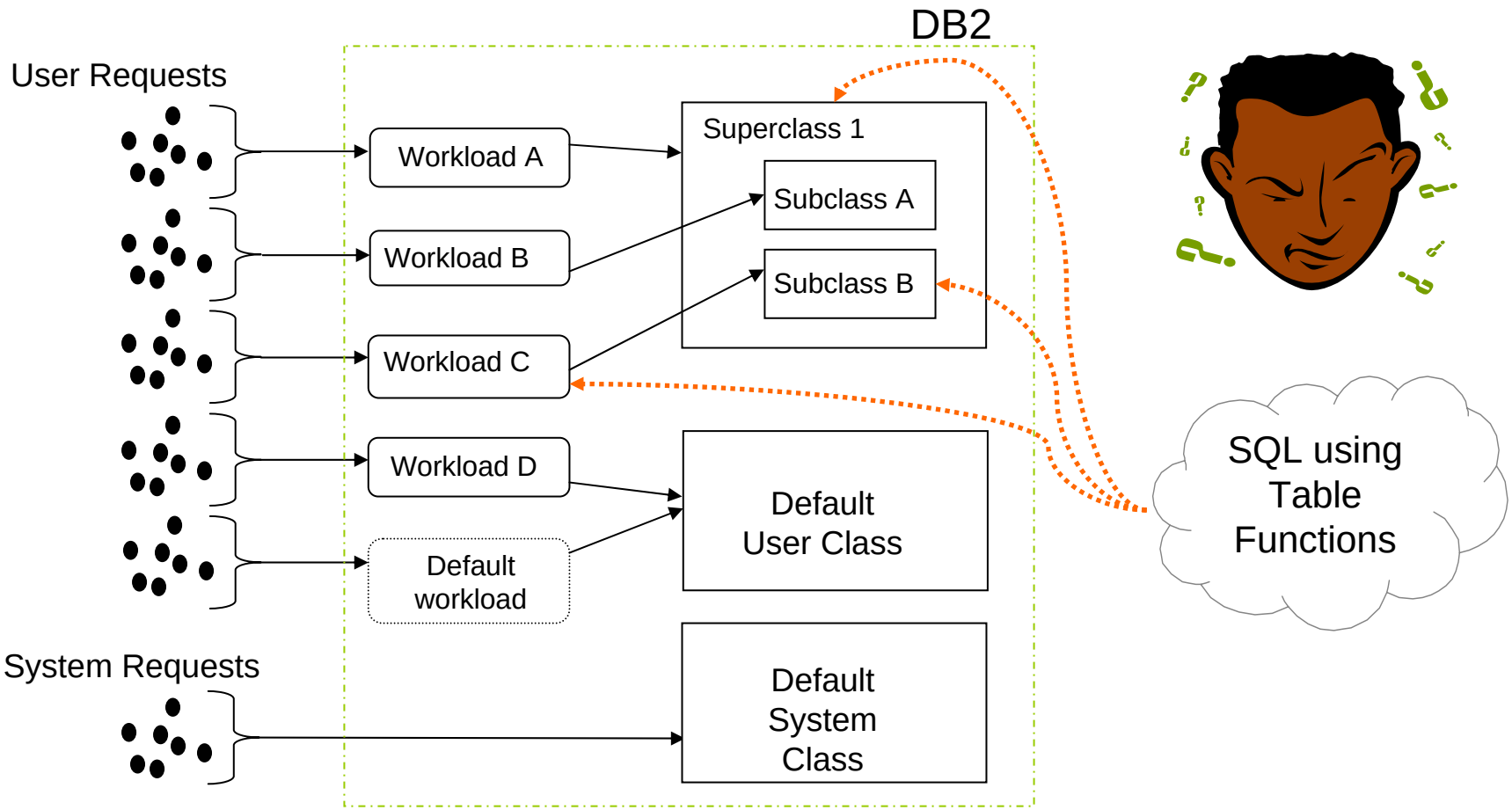
Monitoring Features: Execution Environment

ON DEMAND BUSINESS™

Monitoring the Viper 2 Execution Environment

- Extensive monitoring capabilities are being introduced as part of DB2 Workload Manager in Viper 2
  - New light-weight and pervasive access via SQL
  - Fine granularity of control for collecting this information
- Unprecedented access to detailed information about the execution environment:
  - About active workloads and their activities
  - About active workloads and agents in service classes
  - About statistical information on WLM objects

Ad Hoc Query Support



Access Provided Via Table Functions

- Provide access to “live”, in-memory information on all main WLM objects
- Granular control over the amount of information accessed
- Fast access with minimal impact on DB2 performance
- “Single system” perspective for DB2 with DPF
  - Can see what is happening across all database partitions

Example #1: “Who’s on the system?”

```
select system_auth_id, application_name, workload_name,
        workload_occurrence_state
from
table(wlm_get_service_class_workload_occurrences('','',-1))
```

Output:

```
SYSTEM_AUTH_ID   APPLICATION_NAME   WORKLOAD_NAME                WORKLOAD_OCCURRENCE_STATE
---------------- ------------------ ---------------------------- ------------------------------------------------
PBIRD            db2bp              SYSDEFAULTUSERWORKLOAD       UOWEXEC
NEWTON           db2bp              SYSDEFAULTUSERWORKLOAD       UOWWAIT
PBIRD            db2bp              SYSDEFAULTUSERWORKLOAD       UOWWAIT

  3 record(s) selected.
```

Example #2: “What’s being run on the system?”

```
select application_handle, uow_id, activity_id, activity_type,
        activity_state, local_start_time
from
table(wlm_get_workload_occurrence_activities(cast(null as bigint), -2))
```

Output:

```
APPLICATION_HANDLE UOW_ID      ACTIVITY_ID ACTIVITY_TYPE        ACTIVITY_STATE   LOCAL_START_TIME
---------- ---------- ---------- ---------- ---------- ----------
114                15          1 READ_DML                 EXECUTING        2007-08-12-04.43.09.735069
121                 4          1 READ_DML                 IDLE             2007-08-12-04.27.12.737934
122                 2          1 READ_DML                 IDLE             2007-08-12-01.17.02.425695

  3 record(s) selected.
```

Example #3: “What is that SQL statement?”

```
select name, value
from
table(wlm_get_activity_details(122,2,1,-2))
where name in ('ENTRY_TIME', 'LAST_REFERENCE_TIME',
                'STMT_TEXT')
```

Output:

```
NAME                            VALUE
------------------------------- -------------------------------------------------------------------------------
ENTRY_TIME                      2007-08-12-01.17.02.425695
LAST_REFERENCE_TIME             2007-08-12-04.27.56.079599
STMT_TEXT                       select * from pbird.employee for update

3 record(s) selected.
```

Monitoring Features: Activity information

ON DEMAND BUSINESS™

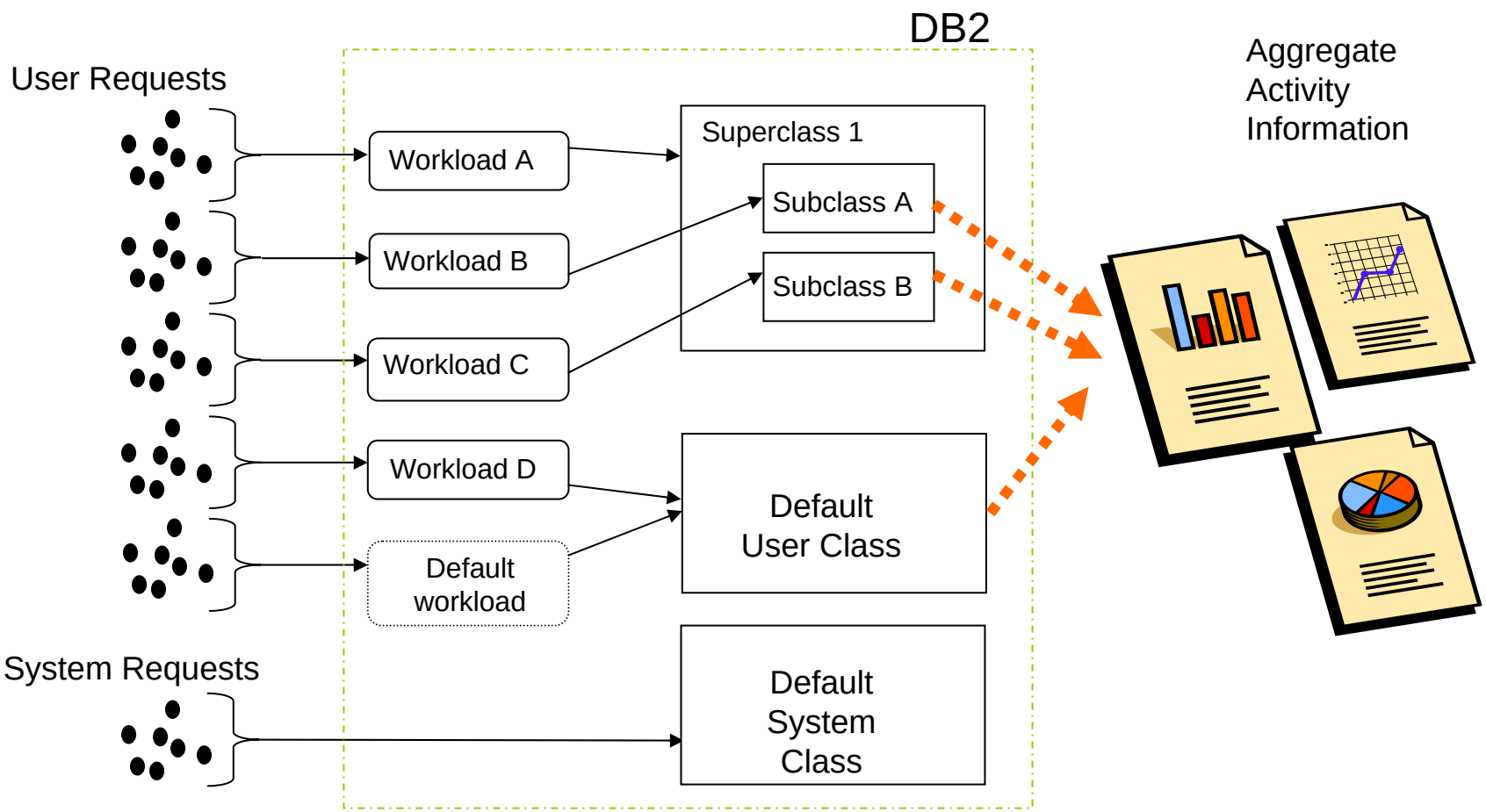
Monitoring Activities in Viper 2

- New Activity event monitors introduced
  - Increased depth of information available for individual activities
  - New types of information available for aggregation of activities in a service class
- Fine granularity of control for collecting this information
  - Individual settings on service class, workloads, thresholds, and more
- Enhanced support provided for common scenarios:
  - Workload profiling
  - Workload capture

Support for Workload Profiling

- New aggregate information available for each service class
  - Basic processing statistics
  - Optional aggregate information
    - Includes distribution information for various activity characteristics such as response time and DML estimated costs
- Gathering can be automated new automated collection mechanism
  - Enables capture of statistics and aggregate activity information to event monitors at regular intervals
  - Can also be pulled manually using WLM_COLLECT_STATS stored procedure

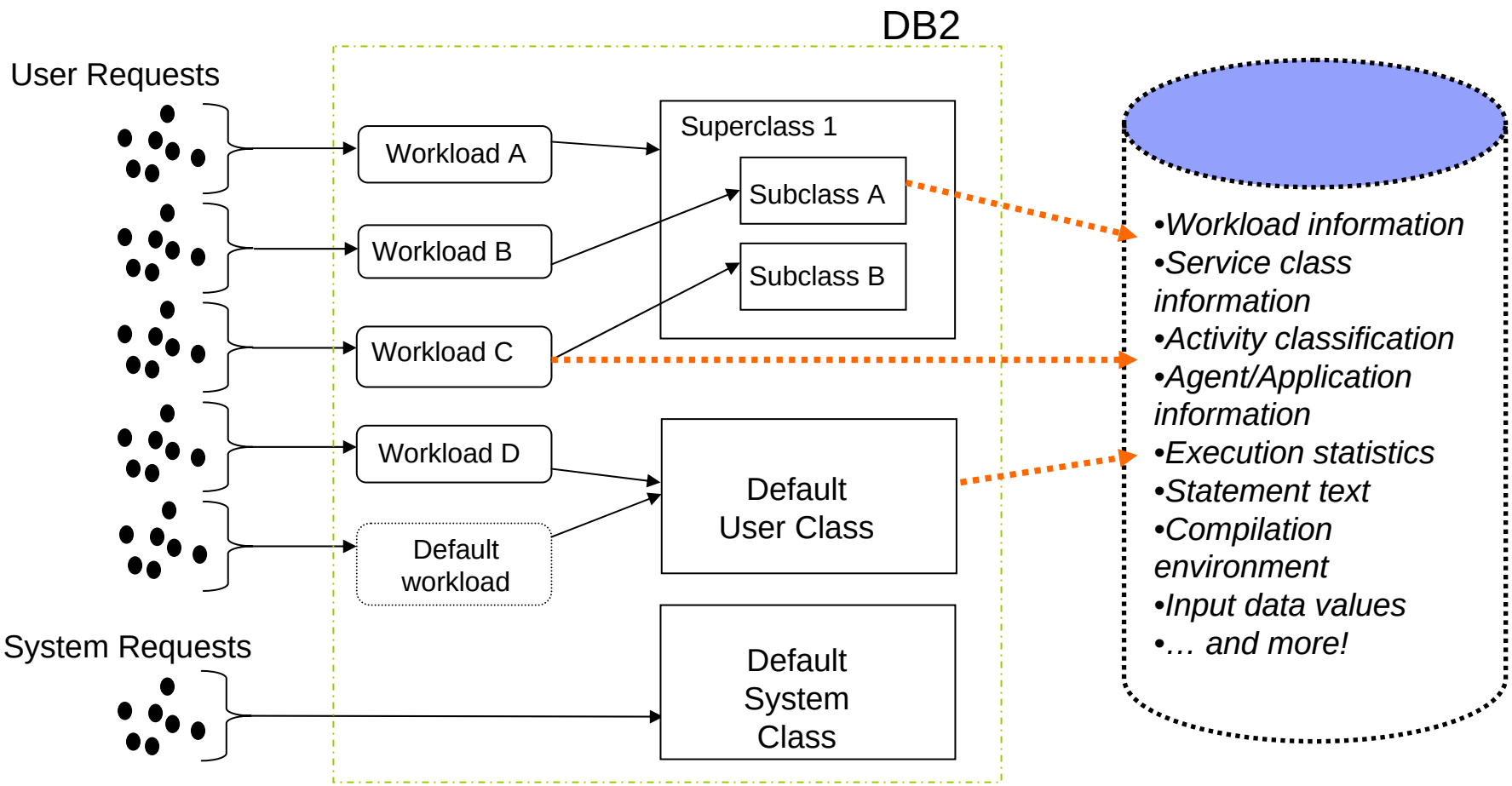
Workload Profiling Support



Support for Workload Capture

- Three levels of information available for individual activities
  - Default, detailed, detailed with input data values
    - Default information includes WLM Identification and basic time statistics
    - Detailed information includes statement text (static and dynamic SQL) and Compilation environment
- Can be activated at different levels and with different settings for each
  - Control settings available at workload, service class, and more
- Can be collected at coordinator node or at all nodes
- Immediate capture of information at a particular point in time enabled via new stored procedure
  - WLM_CAPTURE_ACTIVITY_IN_PROGRESS()
    - Allows for ad hoc capture prior to operator cancellation of activity

Workload Capture Support



Information On Demand

Advanced Features

ON DEMAND BUSINESS™

Introducing the DB2 Threshold

- An automated way to enforce rules or establish limits in the DB2 execution environment
  - “Trigger”-like mechanism based on predefined elements
  - Can be defined at different levels within the database
  - Oversees different aspects within DB2
    - Activities covered include SQL statements and LOAD utility
- Provides support for common scenarios:
  - Controlling “rogue” queries
  - Concurrency control

When a DB2 Threshold is Violated…

- An event is written to the new Threshold Violation event monitor

- DB2 will automatically take the requested actions:
  - Option to collect detailed activity information for later analysis
  - One of the following:
    - Terminate the activity (STOP EXECUTION)
      - An error is returned to the application if STOP EXECUTION is the threshold action
    - Allow it to keep processing (CONTINUE)
      - No error is returned to the application if STOP EXECUTION is the threshold action

Key DB2 Thresholds in Viper 2

- Activities
  - Concurrent DB Coordinator Activities
  - Estimated SQL Cost
  - SQL Rows Returned
  - Activity Total Time
  - SQL System Temp Space
- Workloads
  - Number of Concurrent Workload Occurrences
  - Concurrent Activities in a Workload
- Other
  - Connection Idle Time

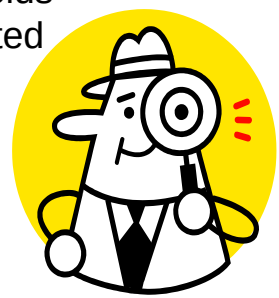
Support for Controlling “Rogue” Queries

- With DB2 thresholds, behavioral norms can be established and enforced on all activities within a threshold’s domain
  - Enables proactive detection and control of problem queries
- Can have multiple thresholds in place at different levels for same activity
  - Define a DB2 Threshold that detects any DML statements with unusually high estimated cost in order to collect information for later analysis but let the work continue
  - Define a DB2 Threshold that detects any SQL statement that has been running longer than 10 minutes, collect information for later analysis, and stop the statement’s execution

Support for Controlling “Rogue” Queries

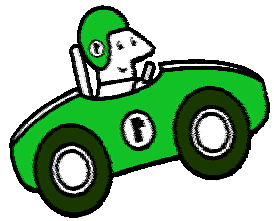
Activity Submitted for Execution

Predictive Thresholds Evaluated



Activity Execution Starts

Activity Execution Ends



Reactive Thresholds Evaluated (ongoing)

Support for Concurrency Control

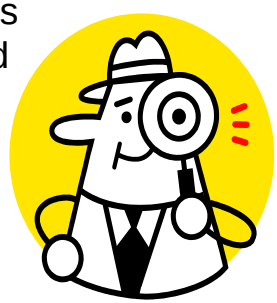
- Some types of DB2 Threshold are “concurrency thresholds”
  - provided to limit and control the maximum number of things allowed to execute at one time
  - Can limit how many can execute concurrently and how many can wait (i.e. queue limit)
    - Threshold is violated when queue limit is reached

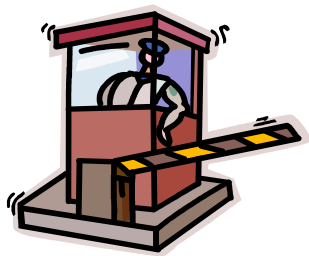
- Using concurrency thresholds
  - Enables control over surges and unexpected peaks in the active workload
  - Allows for control over competing workloads sharing the same execution environment
  - Enables control over concurrency rates for disruptive activities
  - Enables passive profiling and validation of planning assumptions

Support for Concurrency Control

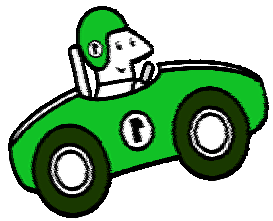
Activity Execution Ends

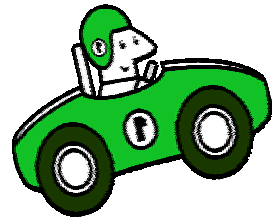
Predictive Thresholds Evaluated





Activity Execution Starts





Activity Submitted for Execution

Concurrency Thresholds Evaluated

Reactive Thresholds Evaluated (ongoing)

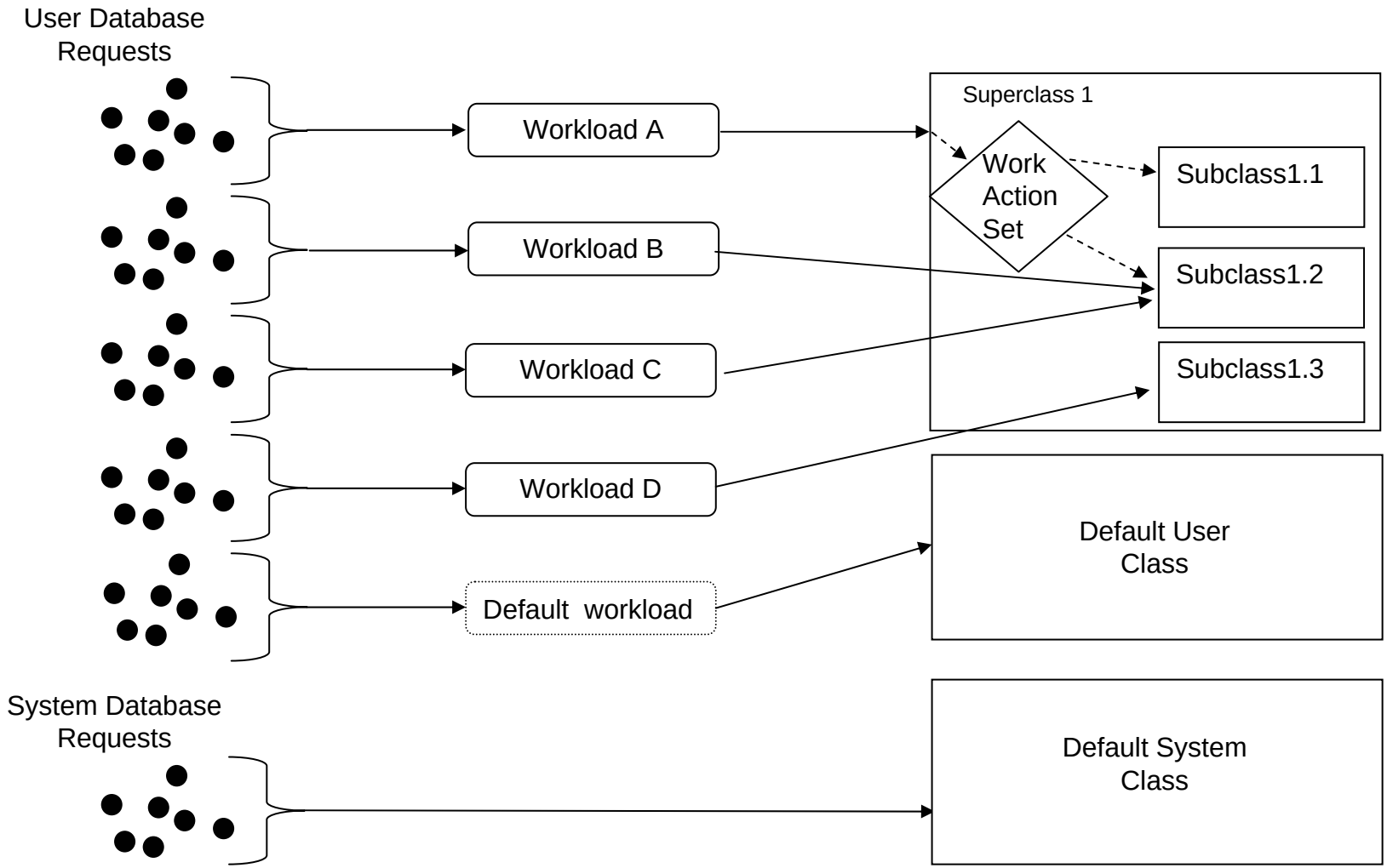
Introducing the Work Action Set

- Sometimes, there is a need to treat activities differently based on either their type or some other individual characteristic
  - "Put DML in a different service class than DDL"
  - "Put all Read queries of less than 100 timerons in a different service sub class than all the other Read queries"
  - "Put a restriction on the number of concurrent Loads allowed to execute on the database at any one time"
- The DB2 Work Action Set provides this capability
  - Can be used at the database level to apply DB2 Thresholds with discrimination
  - Can be used at the level of a DB2 service super class to map to subclasses with discrimination

Possible Actions within a DB2 Work Action Set

- Everywhere
  - COUNT ACTIVITY
  - PREVENT EXECUTION
  - COLLECT ACTIVITY DATA
- Database
  - Apply a DB2 Threshold
- Service Super Class
  - Map work to a different service sub class (within same super class)
  - COLLECT AGGREGATE ACTIVITY DATA

Example of Work Action Set Mapping Concept



Query Patroller and Workload Manager

ON DEMAND BUSINESS™

Query Patroller in DB2 Viper 2

- Still supported and still works as it does in DB2 9.1
- No longer strategic for DB2 workload management
- Is only aware of work executing in the Default User Service Class
- Is not aware of work in other service classes

QP and Viper 2 WLM: Essential Differences

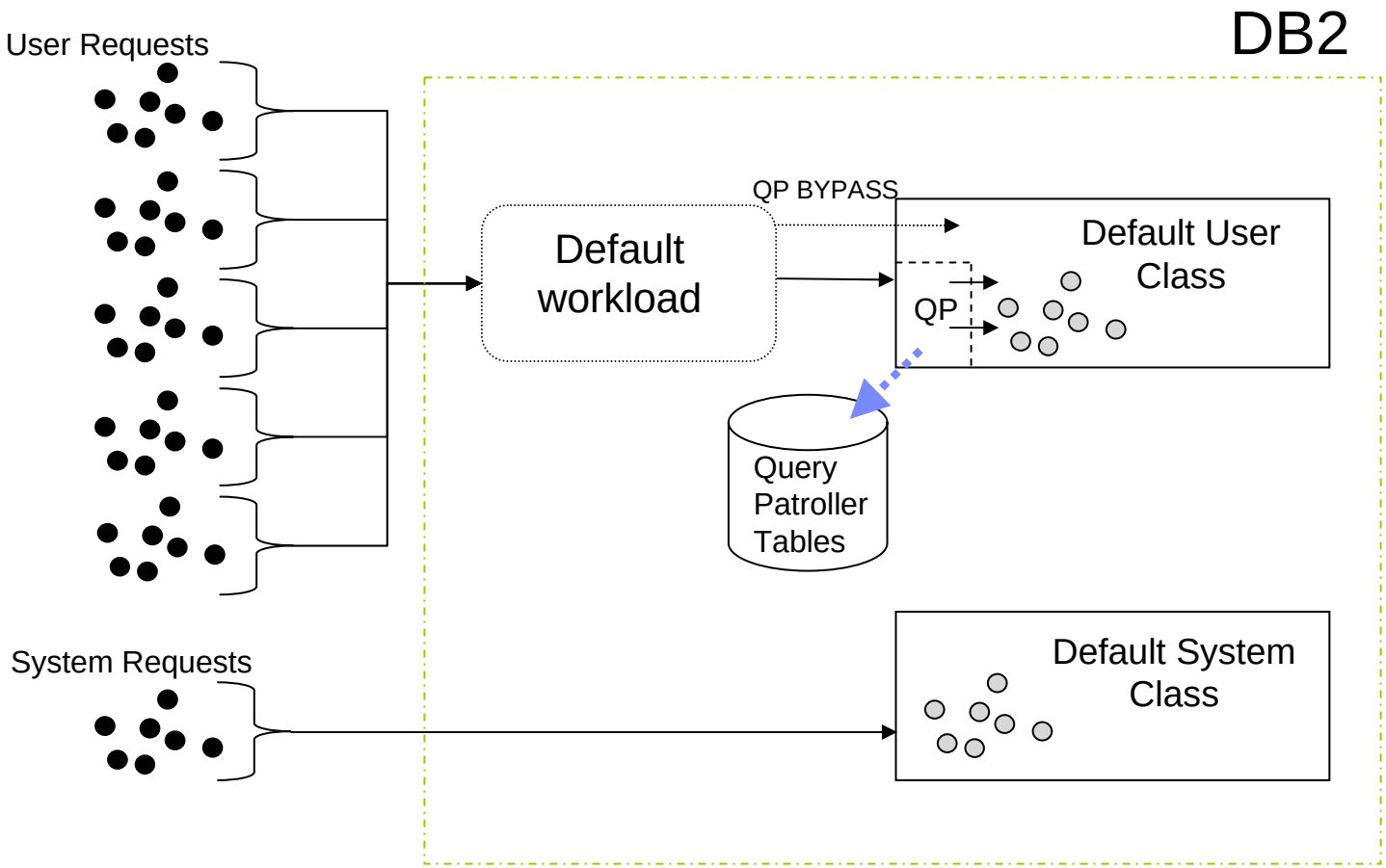
Query Patroller

- acts as a “doorman”: once work is admitted, it is free to execute as it desires
- has no further information on the submitted work until execution has completed
- is only aware of the coordinator perspective for submitted work
- does not provide any mechanism to explicitly control the resources used for execution
- details on all managed activities written to control tables (disk)

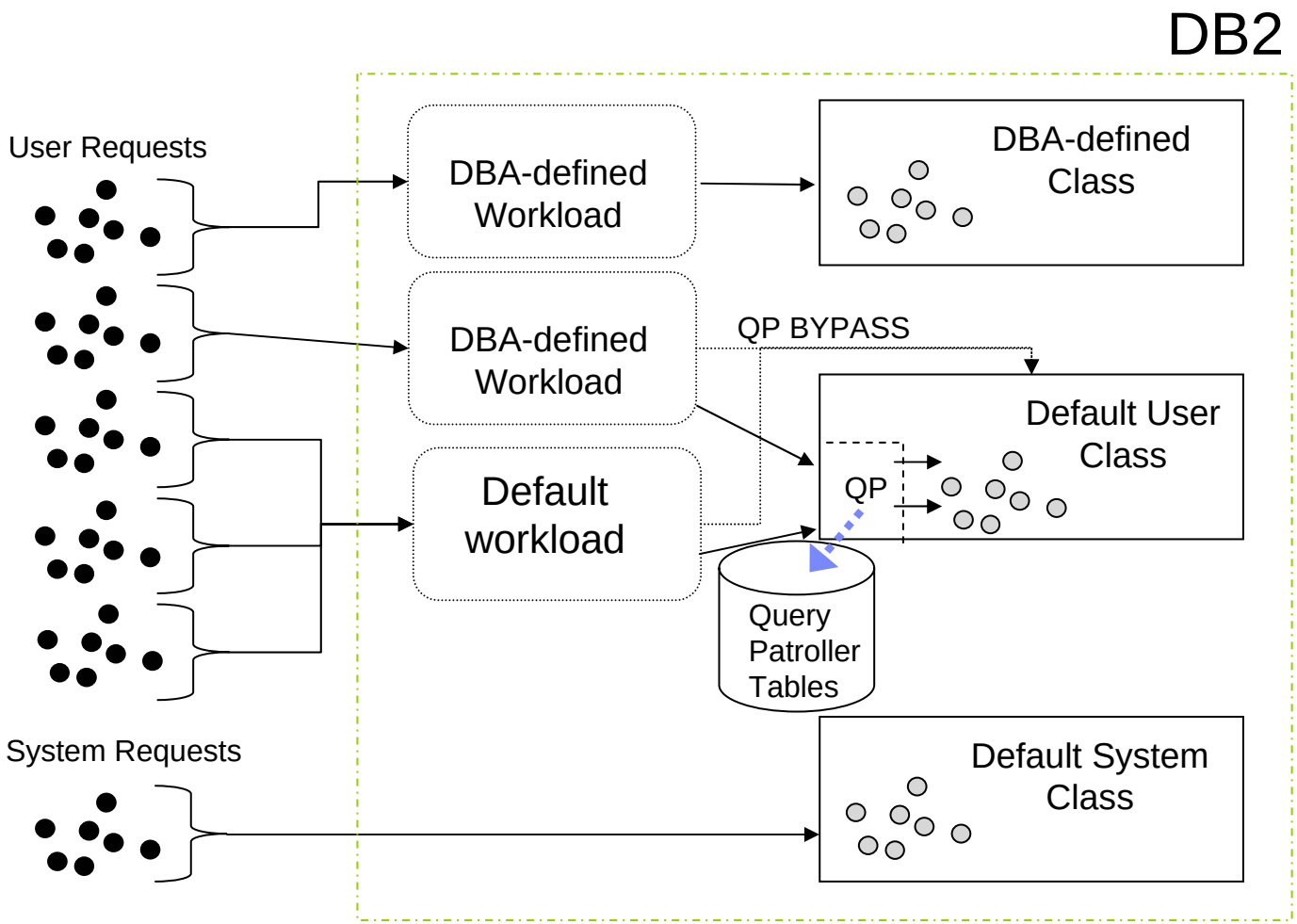
Viper 2 WLM

- acts as a ‘hall monitor”: it ensures that work goes to its correct place and follows the relevant rules during execution
- can show the current state of the application, SQL, and even agents at any time during execution
- is aware of the work across all database partitions
- provides mechanisms to control and influence resources used during execution
- nothing gets written to disk unless requested by user

QP within the DB2 Viper 2 Execution Environment



DB2 Viper 2 Execution Environment (Coexistence)



Summary

- Viper 2 introduces an important growth in the workload management capabilities of DB2 for LUW:
  - Better identification and control of applications (DB2 Workload)
  - Direct administrator manipulation and control of the DB2 execution environment including resource allocation (DB2 Service Class)
    - Tight integration with AIX WLM
  - New mechanisms to aid in the detection and control of “rogue” queries as well as to implement concurrency control (DB2 Threshold)
  - Advanced monitoring capabilities of the execution environment to support ad hoc monitoring as well as workload profiling and capture

Questions?

Paul Bird
pbird@ca.ibm.com

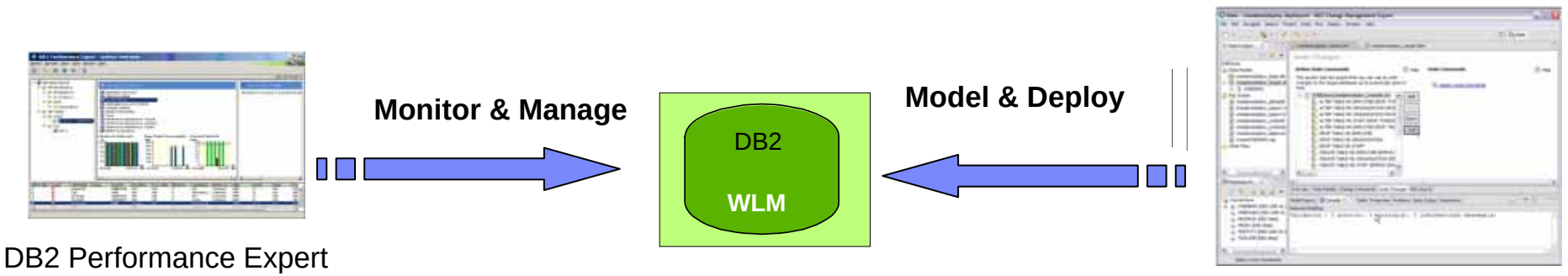
ON DEMAND BUSINESS™

Supplemental Slides

ON DEMAND BUSINESS™

DB2 9.5 WLM Tooling in DB2 Warehouse

- DB2 Warehouse Enterprise **includes** DB2 PE
- **Design Studio** for WLM Setup
  - **Eclipse** based WLM **modeling**
  - WLM **deployment** support
- DB2 PE monitors & maintains **history** of
  - WLM **statistics**
  - WLM **Setup**

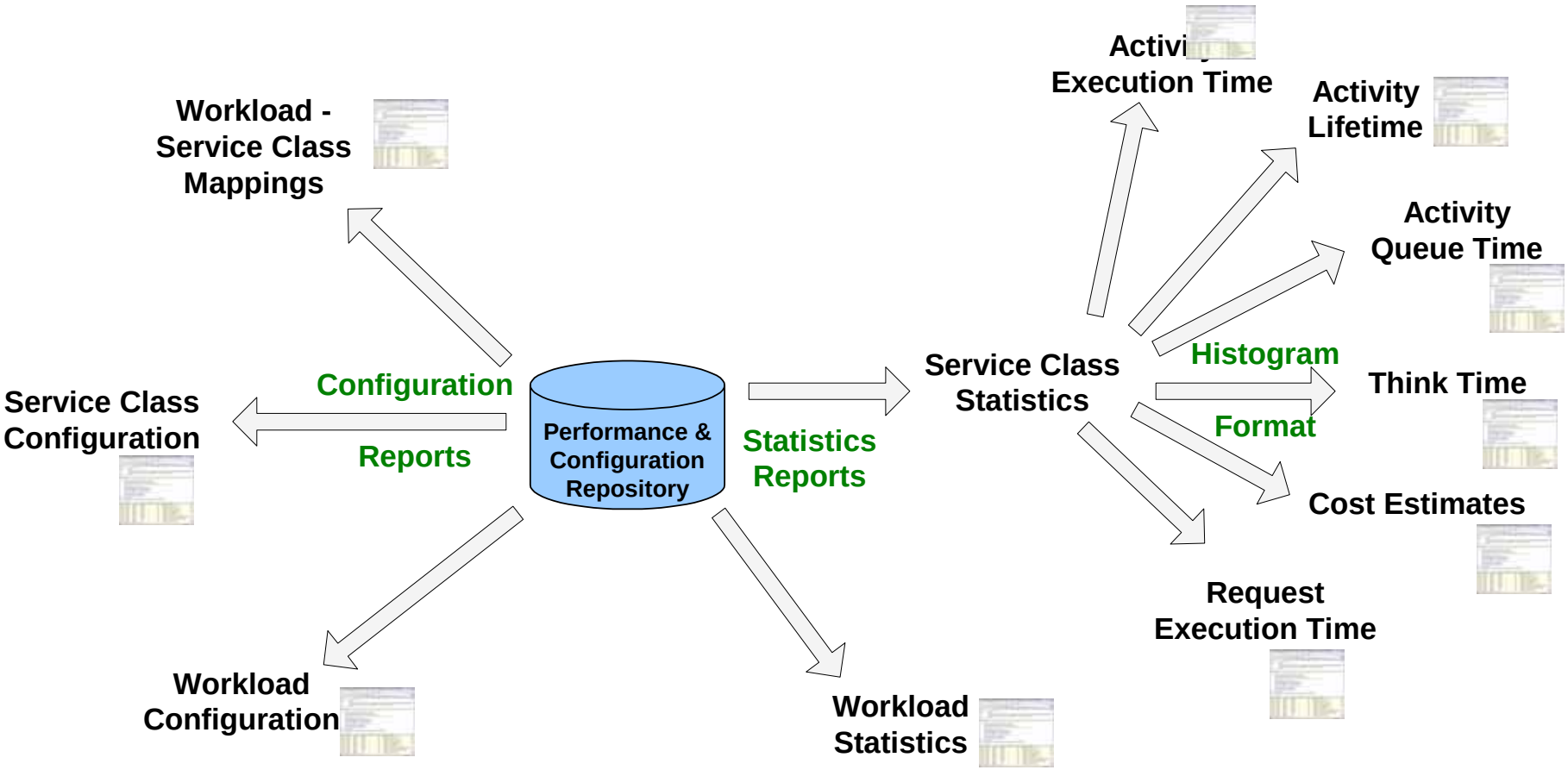


DB2 PE & DB2 9.5 WLM

- **Monitor** service classes, workloads & work classes
  - Comprises sophisticated usage **statistics** & **definitions**
  - Real-time & history
- **Understand** workload at a point in time
- Understand **workload patterns** & long term development
  - Derive effective service classes & thresholds
- Workload analysis for
  - Workload profiling
  - Accounting

WLM Reports

- SQL-based reports on long-term WLM **configuration** & **statistics**
  - ➔ Extensible & customizable by the user



WLM Support in Design Studio (1 of 2)

1. Reverse engineering
   - Pull WLM solution from a database into Design Studio
2. Guided creation of WLM solutions
   - Resource Sharing
   - Enforcing Limits for Activities
   - Enforcing Limits for Concurrent Activities
3. Simplified viewing/manipulation of WLM entities
   - Tree view – hierarchical overview of WLM solution
   - Properties views

WLM Support in Design Studio (2 of 2)

4. Validation
   - Pieces not linked up
   - Workloads effectively disabled because no authority granted
5. Robust execution
   - Minimal DDL to get you from existing -> desired scheme
   - Generate and/or execute DDL
   - Error checking to recover from errors