WP3 F4K Project

Final Year Review Presentation

Jessica Chen-Burger
Gaya Nadarajan
Cheng-Lin Yang
University of Edinburgh
Nov 29, 2013
Luxembourg

Goals for the Workflow System

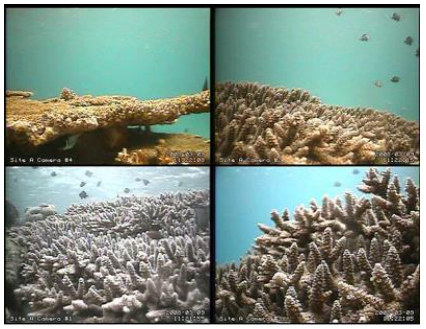
- To create a working workflow system for viewing and analysing fish videos
- The workflow system
  - load videos captured by F4K partner NCHC from open sea;
  - analyses and processes them using VIP modules generated by F4K teams, to run them on NCHC HPC machines;
  - Supports user queries via F4K UI.

3 Years Ago….

We have promised

- WP 3.1 – understand and formally describe the multi-disciplinary and interdependent problem domains
  - Deliverables: formal domain descriptions (inc. ontologies)
- WP 3.2 - design to cope w/ the integration of on-going F4K teams research efforts
  - Deliverables: System design
- WP 3.3 – Workflow Implementation

A high level view of the use of the workflow system



System initiated Batch request

Ontologies & Models

videos

annotations

Admin

Initial Batch run request

NCHC

User query

Marine Biologists

Analysed results

Workflow System

Select and run modules

Send executables

results

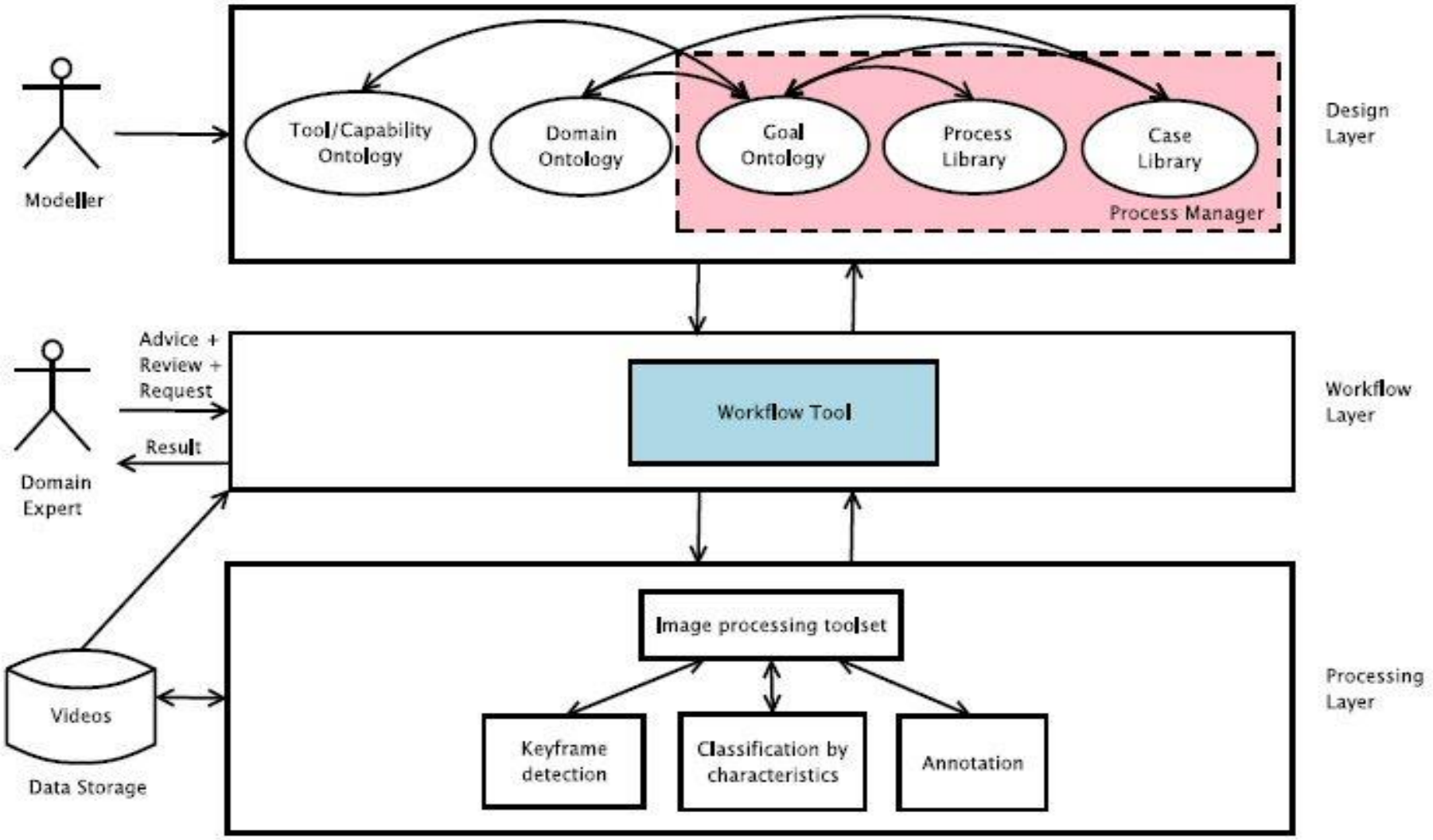
results

WP 3.1 Achievements

- A set of brand new domain ontologies were built
  - Goal ontologies
  - VIP modules capabilities ontologies
  - Video description ontologies
  - Partial process model in libraries
  - Extensions to the EU FAO fish ontologies
    - 27 new fish identified and added
    - 6 new object properties identified and added

WP 3.2 Achievements:

Semantics based, Decoupling, re-Configurable Framework



SWAV – Semantics based Workflow tool for Analysing Videos

Select video 28.mpeg

- Classify video according to brightness, clearness and algal levels
- Detect and count fish in each frame of video
- (•) Classify video, detect and count fish in each frame

Results dislayed. Ready. Select a video and goal to proceed.

Video Descriptions (Optional)

Brightness

dark medium bright

Clearness

blur medium clear

Algal level

green not green

Constraints (Optional)

Performance

fast less memory

Quality

reliable robust

Unknown objects

ignore score

Reset GO

First version of system prototype

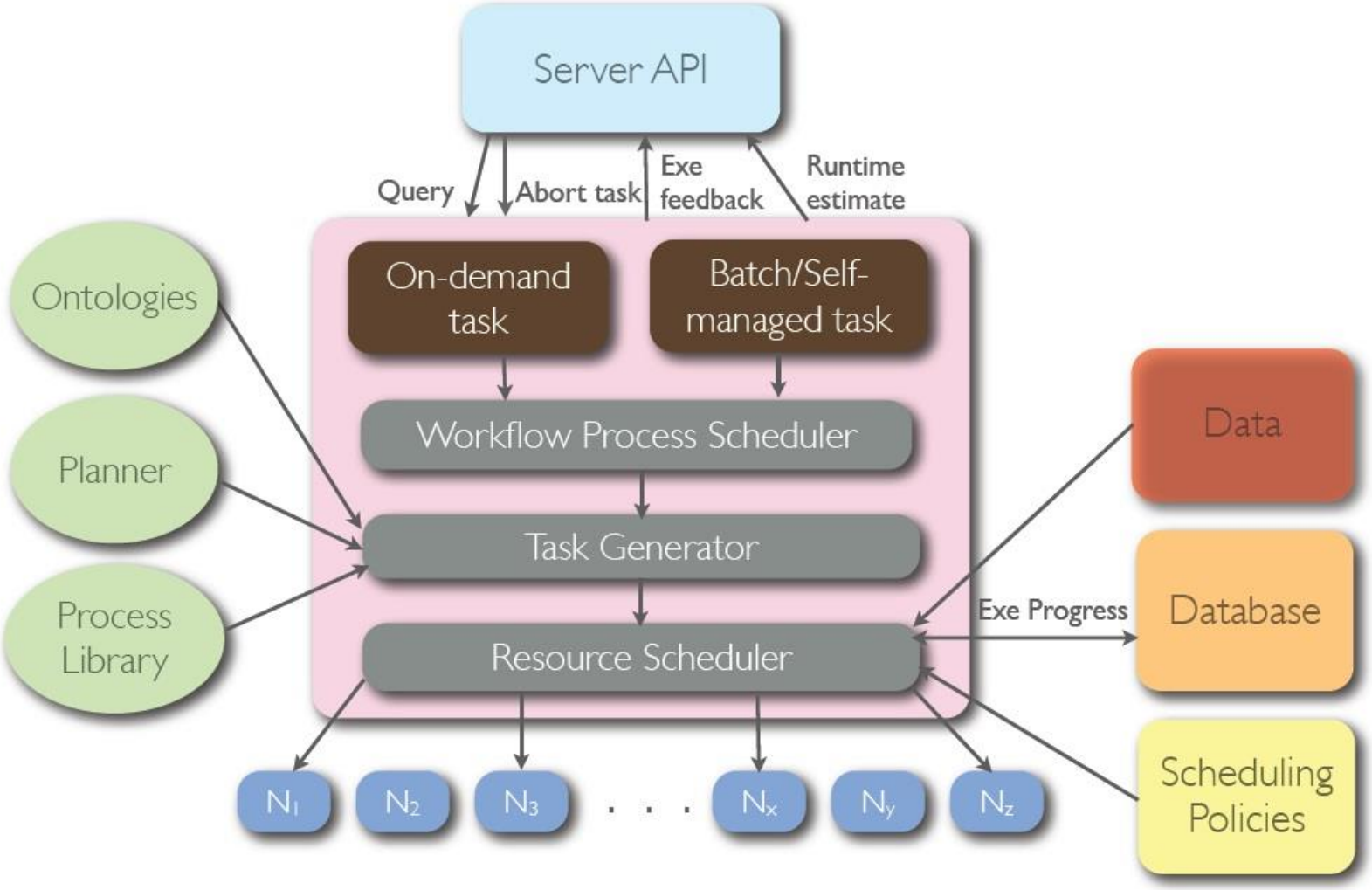
WP 3.3 Achievements System Implementation

- Enhance of the workflow system
- Development and run workflow system in the NCHC HPC environment
- Link up with new, continuously developing VIP, UI F4K modules
- Run in a continuously improving HPC environment of NCHC

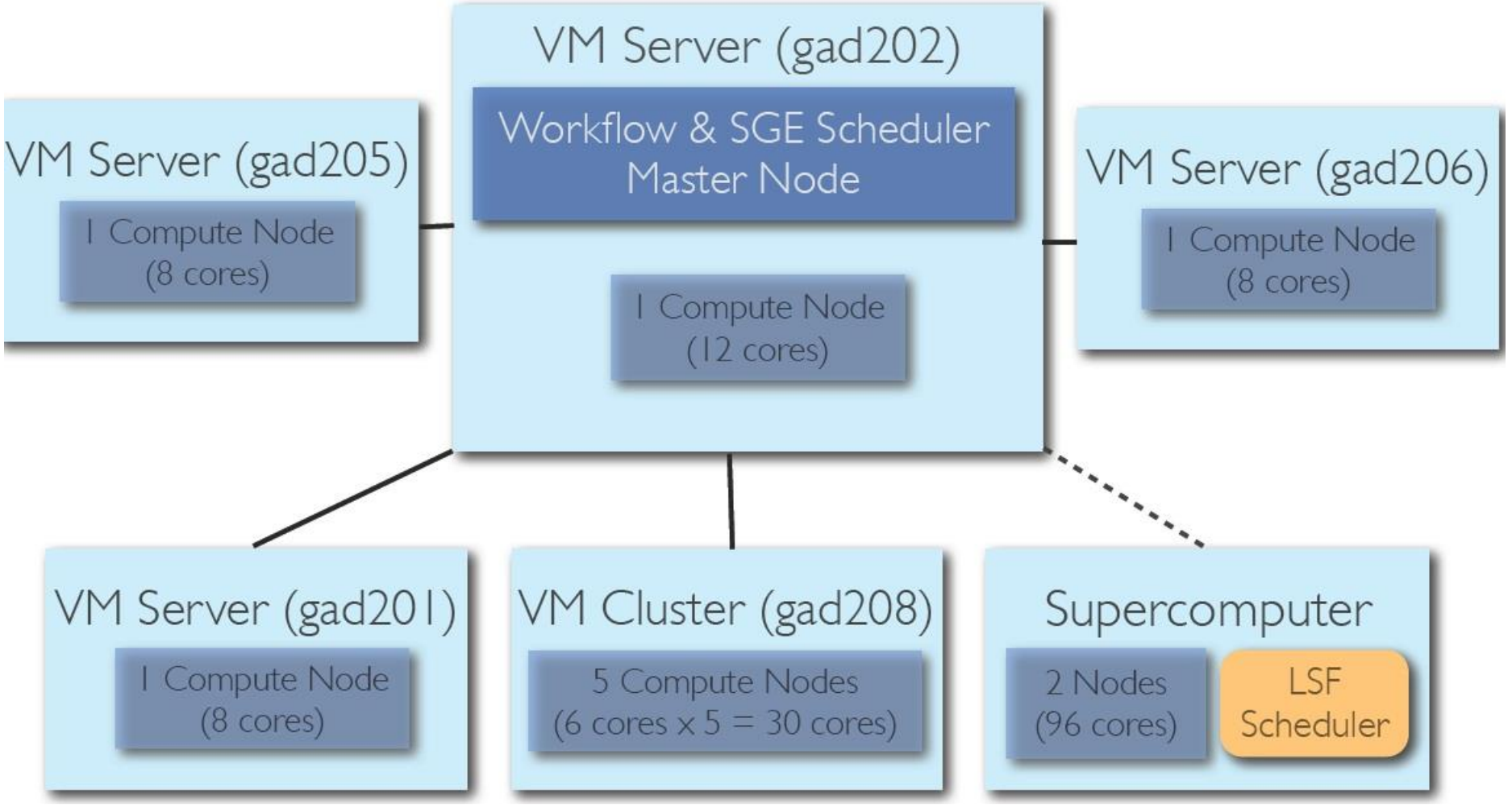
WP 3.3 Achievements: Virtual Workflow Machine

- Dynamic composition of abstract workflow based on (batch/user) requests
- Create concrete workflow
  - instantiation abstract workflow with software modules w/ appropriate parameters
- Execute concrete workflow – spawn run-time execution jobs in HPC environment
  - Allocate jobs to machines queues for execution
- Store job execution status & results in suitable databases

WORKFLOW MANAGER ARCHITECTURE



COMPUTE ENVIRONMENT FOR WORKFLOW



SEVENTH FRAMEWORK PROGRAMME

THE UNIVERSITY OF EDINBURGH

fish 4 knowledge

Support User Queries

- Formally define machine interface with F4K UI module
- Communication, collaboration and integration via formally defined data
- Link up with UI module to support end-to-end F4K user query system operations
- Enable automated F4K system production run

Query Management Table

Field	key	updated by	Data Type	Field Description
query _ID_by_WF	primary key	Database	integer	This is automatically generated and incremented by the database when CWI inserts a new query
query_content		CWI	string	query content is the task to be executed: inc. detect_track_fish, recognise_fish, execution_progress, abort.
generated_by		CWI/WF	integer	This indicates who generated the query, 1 indicates CWI and 0 indicates WF
data_start_date		CWI	date	Please provide a start date for the video to be processed, format is yyyy-mm-dd.
data_start_time		CWI	time	When not provided, the default is 06:00:00. Otherwise, please provide a start time for the video to be processed, e.g. 06:00:00 is 6am, and 14:00:00 is 2pm, etc. Please provide in hh:mm:ss
data_end_date		CWI	string	Please provide an end date for the video to be processed, format is yyyy-mm-dd.
data_end_time		CWI	date	When not provided, the default is 17:50:00. Otherwise, please provide a start time for the video to be processed, e.g. 06:00:00 is 6am, and 14:00:00 is 2pm, etc. Please provide in hh:mm:ss
video_location_videoNumber		CWI	time	This field indicates where the videos are shot. This is based on a combination of fields names of the 'cameras' table. These fields are called "Location" and "Video Number". Valid values are NPP-3/1, NPP-3/2, NPP-3/3, NPP-3/4, HoBiHu/1, HoBiHu/2, HoBiHu/3, LanYu/1, LanYu/2. However, the user can indicate more than one location/camera in one query. Therefore a valid query may also be, e.g. 'NPP-3/1,NPP-3/2,NPP-3/3,NPP-3/4' (with comma, no space between them).
component_id_detection	(this is an optional field)	CWI		This field allows CWI to specify which component ID to use. Currently, the user can specify the detection component ID. Please see the task mapping for task definition and useage for reference.
component_id_recognition	(this is an optional field)	CWI	integer	This field allows CWI to specify which component ID to use. Currently, the user can specify the recognition component ID. Please see the task mapping for task definition and useage for reference.
query_insert_time_by_CWI		CWI	time stamp	this field is updated when CWI made an insertion of a new query into the DB.
query_start_process_time_by_WF		WF engine	time stamp	this field is updated when WF engine picks up this query and start processing it.

Job and Queue Management

- Queue selection for jobs submission
  - Based on capabilities and availabilities (job submission patterns) (Wind Rider)
  - Based on stability of queue (VM)
- F4K system performance analysis: Speed, Robustness
  - Comparison of individual modules
  - Comparison of different configurations
  - Comparison of data of varying qualities
    - normal, bad weather (typhoon) and perfect video data (manual captured)
- Continuous run-time estimate for job execution

Individual Module Performance Analysis

Time: in seconds per video

component_id	average execution time	average queuing time	max execution time	min execution time	average DB write time	number of videos used
128 (recognition)	8796 (~2.4 hrs)	6164 (~1.7 hrs)	355381 (~ 4 days)	15	68	63557
135* (detection)	734 (~12 mins)	90 (1.5 mins)	19604 (5.4 hrs)	0	93	363350
136* (recognition)	9902 (~2.75 hrs)	42655 (~ 11.5 hrs)	344113 (~ 4 hrs)	16	32	25849
141 (detection)	892 (~14.7 mins)	31460 (~ 8.7 hrs)	2845 (~ 47 mins)	10	4	451
142 (detection)	11336 (~3.15 hrs)	53205 (~ 14.8 hrs)	28107 (~ 7.8 hrs)	180	11	443

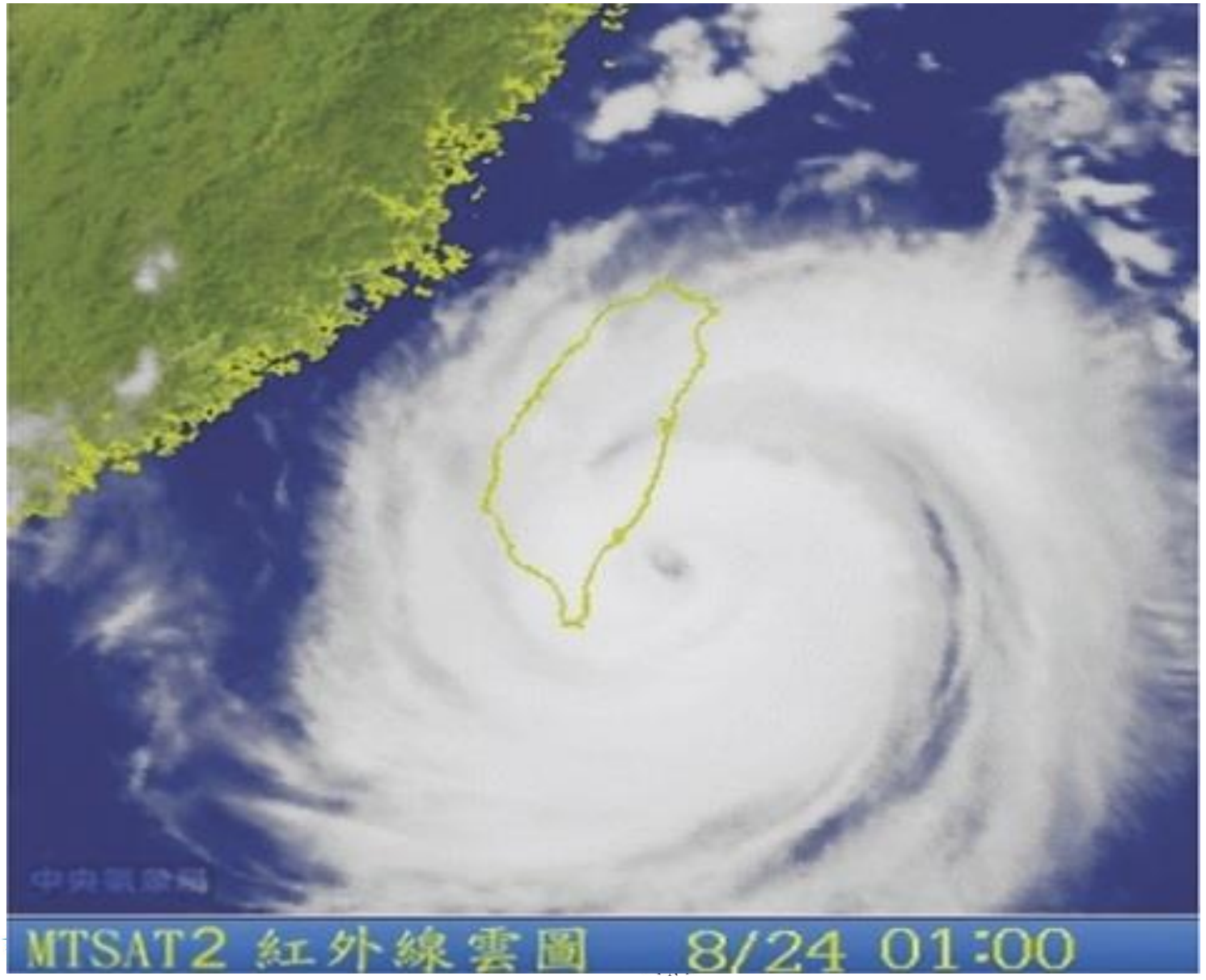
* Default component

Performance Comparison of Different Configurations

cross over with

Different Video Events (Typhoon, Manual Capture)

Typhoon Tembin



Typhoon Tembin (24th August 2012)

3024 videos

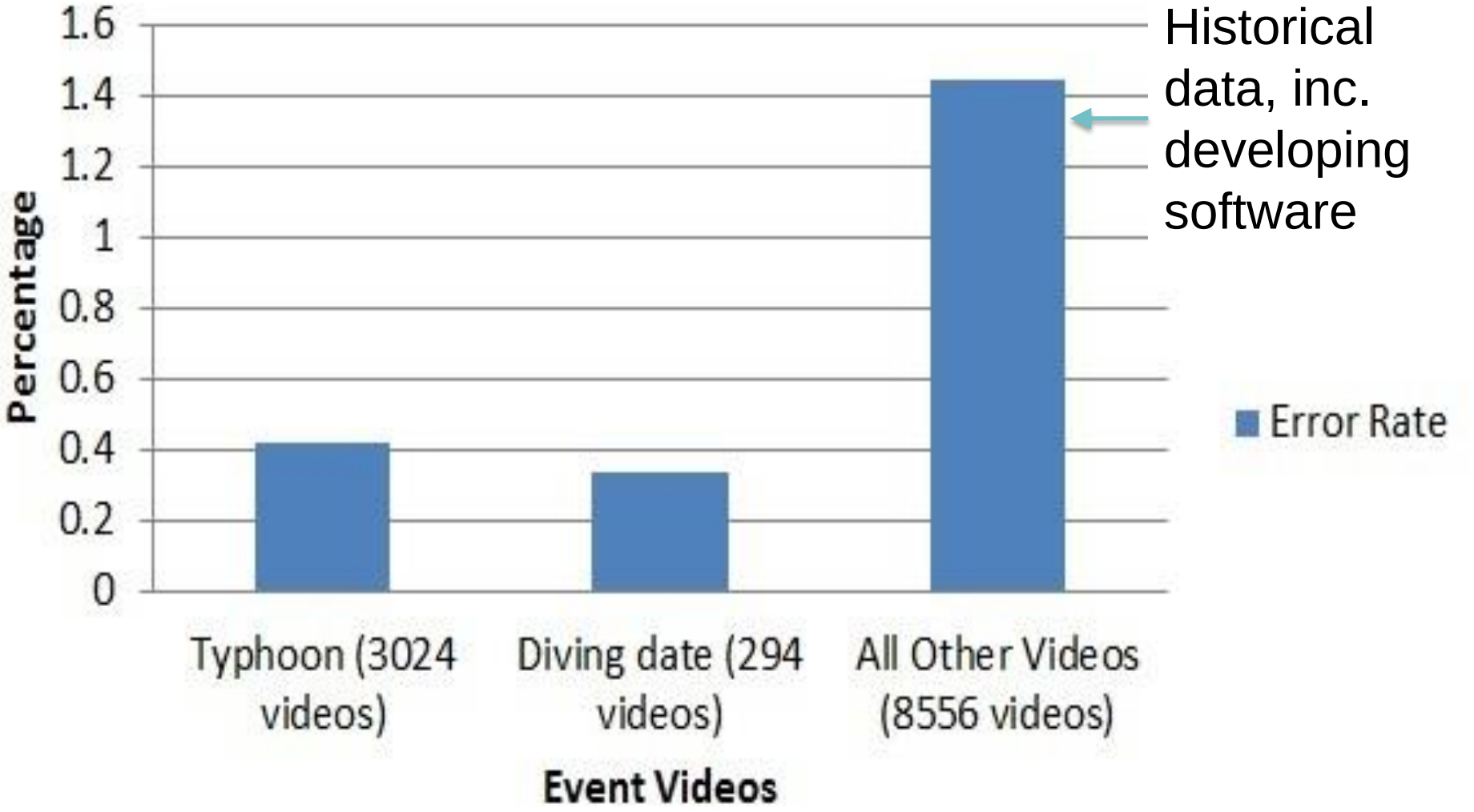
Combination	% of SW Error	% of Missing Video
135-128	0.55%	35.2%
141-128	0.22%	35.2%
142-128	0.44%	35.2%
141-136*	0%	35.2%
142-136*	0%	35.2%
135-136	0%	35.2%

Diving (5th February 2013)

24 videos

Combination	% of SW Error	% of Missing Video
135-128	0%	0%
141-128	0%	0%
142-128	0.17%	0%
141-136	0%	0%
142-136	0%	0%
135-136	0%	0%

Software Execution Error Rate



Historical data, inc. developing software

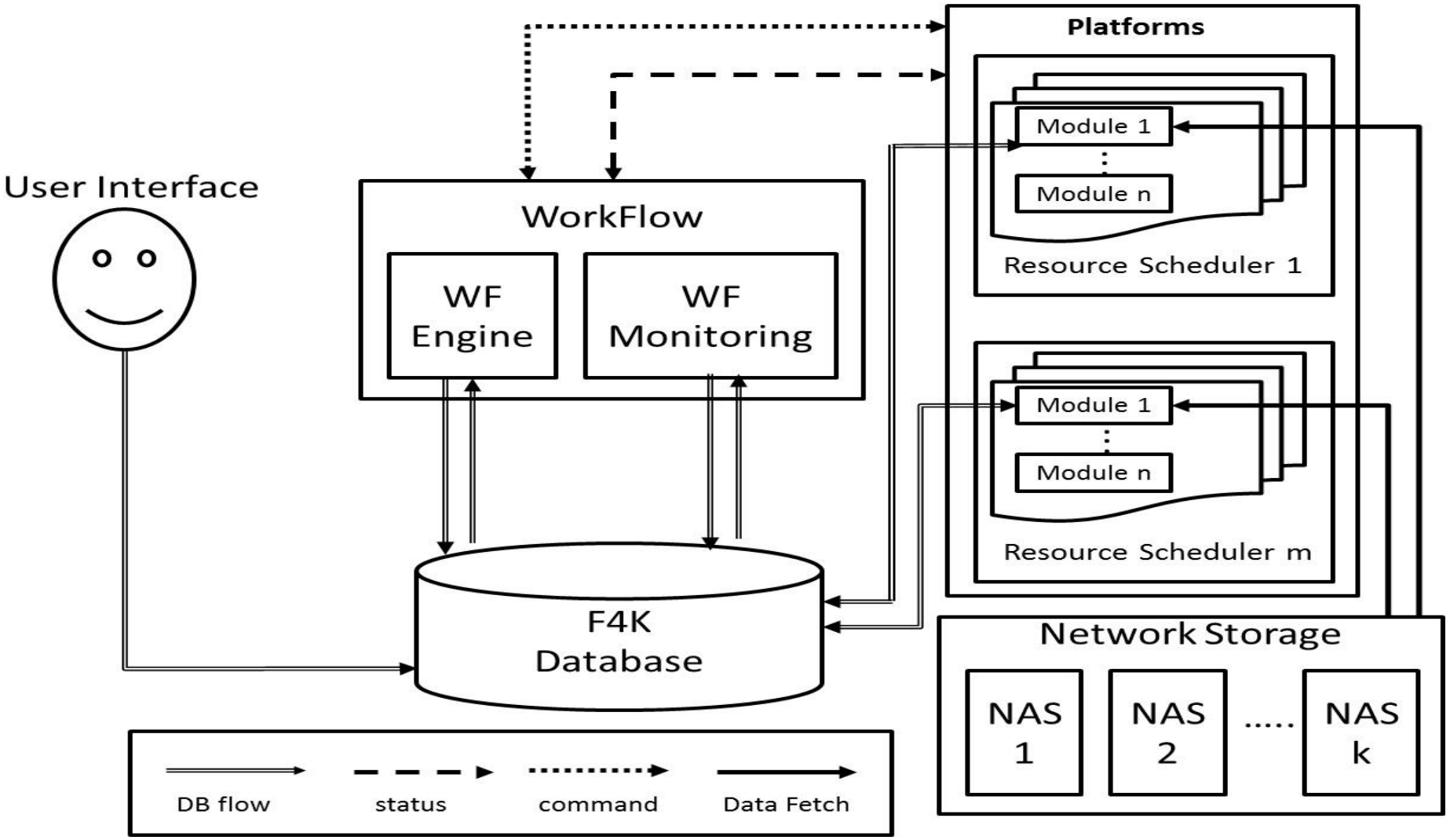
Run-time Challenges

- Failed jobs
  - Failed machines
  - ill-configured machines; lost of library modules
  - Lost jobs (by machine)
  - Database transaction/sharing failure/conflict
  - Forever running/waiting jobs (but no progress made)
  - Software errors
- Machine hung
  - Network problems: UK/Taiwan, NCHC internal network
  - Failed jobs (caused by self/other user jobs/external malware attacks)
- Missing/corrupted (video) data
- Bottleneck at database access
- Job cancellation: user cancellation/system abort
- Scheduled/un-scheduled machine maintenances

What we have not promised to do, but have done

- Job Monitoring and Error Detection
  - Continuously close monitoring and keep track of execution status
- Error Type Definition and Description
  - Queue error (e.g. node down)
  - Failed job (e.g. segmentation fault)
  - Dependency propagation
  - Queuing too long/Failed to terminate (running too long)
  - Suspended jobs (un-expected)
- Knowledge and heuristic based Error Handling Algorithms and Implementation

Workflow Architecture w/ Error Handler



Contribution of Error Handler

Error Type	No Error Handling	w/ Error Handling
Queue error	Wait sysadm for fixing	Automatically resubmit to another queue
Failed job	Non-recoverable	Automatic resubmit
Dependency propagation	Dependent job starves in queue	Automatically handled based on dependee status
Queuing too long	Keep waiting in queue	Automatically increase job priority
Failed to terminate (running too long)	Keep running (waste resource)	Dynamic running time threshold
Suspended jobs (un-expected)	Resume manually until spotted by the user	Automatically resume

Additional Performance Analysis Framework

- Collaborators:
  - Prof. Omer Rana, Cardiff Univ., UK
  - Dr. Rafael Tolosana, Universidad de Zaragoza, Spain
- Quality of Resilience Framework
  - Task composition
  - Resource demanded
  - Module performance breakdown

Analysing QoR for F4K Workflows

Quantitative QoR Metrics Classification: QoR_U: task level							
metric	description	$t1_1$	$t1_2$	$t1_3$	$t1_4$	$t2_1$	$t2_2$
m1	number of alt. tasks	3	3	3	3	1	1
m2	number of input tasks	0	0	0	0	1	1
m3	number of resources	1	1	1	1	1	1
m4	task failure rate	3.02	4.12	6.0	2.1	21.7	12.3
m5	task exec. time (secs)	397	411	1596	1342	4984	13134

Table II
F4K QUALITY OF RESILIENCE METRICS AT TASK-LEVEL

Quantitative QoR Metrics Classification: QoR_U: workflow-level									
metric	description	wf_1	wf_2	wf_3	wf_4	wf_5	wf_6	wf_7	wf_8
m6	avg. number of alt./tasks	2	2	2	2	2	2	2	2
m8	number of task joins	1	1	1	1	1	1	1	1
m9	wf failure rate	12.36%	7.66%	12.9%	8.21%	13.85%	9.15%	11.9%	7.2%
m10	wf exec. time (secs)	2550	5244	5371	4747	7857	9726	5531	14643
m11	overall number of resources	2	2	2	2	2	2	2	2

Table III
F4K COMPILATION OF QUALITY OF RESILIENCE METRICS AT WORKFLOW LEVEL

Test data: using a combination of 4 detection algorithms and 2 recognition algorithms

t1: detection task

t2: recognition task

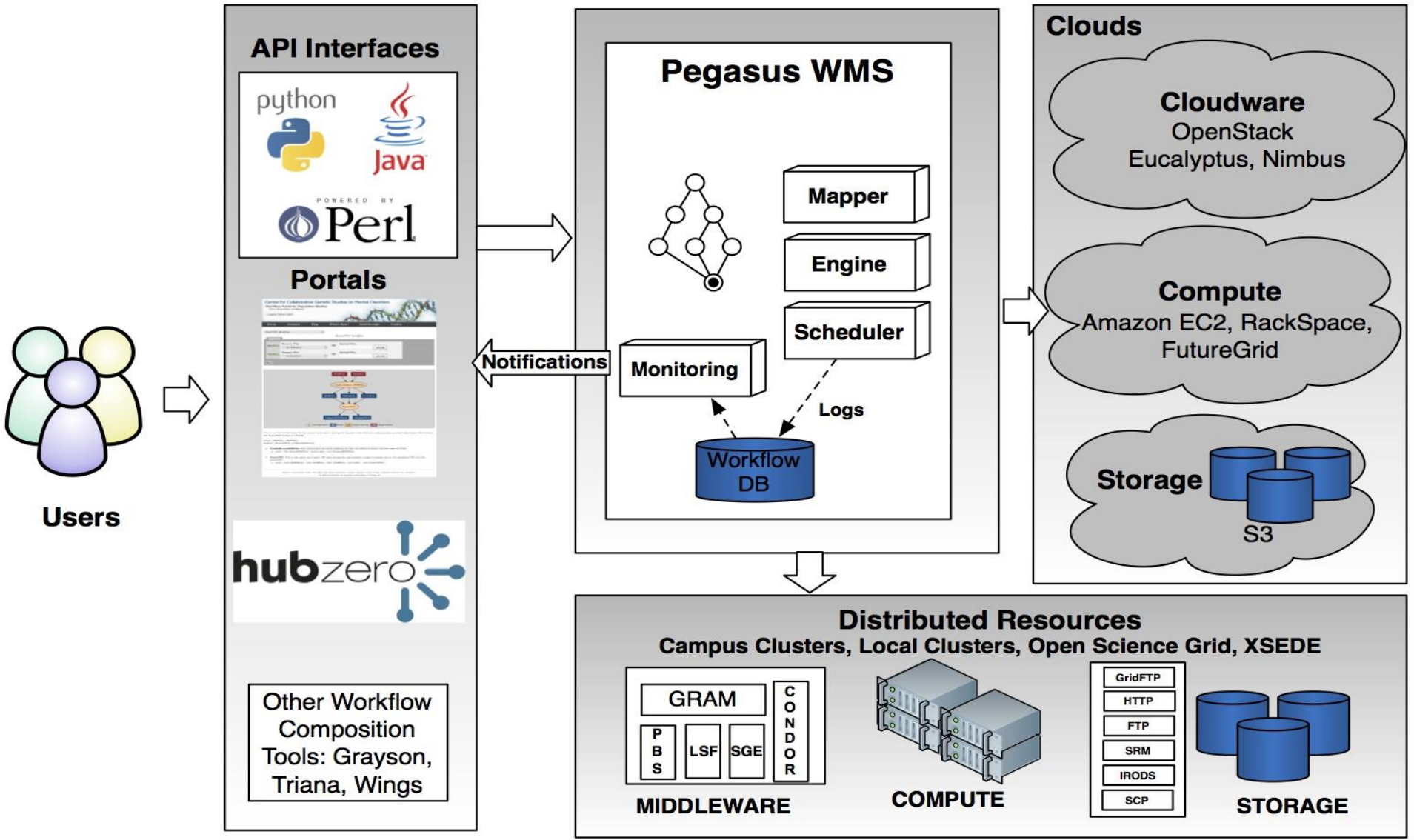
wf1: t11,t21 wf2: t11,t22 wf3: t12,t21 ... wf8: t14,t22

m9: likelihood of a certain workflow instance (combination) failing

[Submitted to *CloudAM 2013*]

Related Work

eScience workflow system: Pegasus



Components in Pegasus

- Mapper (Pegasus Mapper):
  - create abstract workflow
- Execution Engine (DAGMan):
  - meta-scheduler for HTCondor
  - Execute workflow
- Task manager (Condor Schedd):
  - supervise execution
- Monitoring Component (Pegasus Monitord): notify user execution status

Other Related Work

- Pegasus
  - Job submission
  - Execution feedback (successful, failed, suspended)
- Triana
  - Job submission
- Taverna
  - Job submission

Our Scientific Contributions 1

- New ontologies and vocabularies
  - Ontology that translates and maps user goals to operation and task goals
  - Extension to EU FAO fish ontology
- Flexible ontologies and knowledge based virtual workflow framework – decoupling of different domains
- Reconfigurable virtual workflow machine for dynamic workflow composition and execution
- Knowledge and data based integration approach
- Realisation of workflow execution and management
  - From abstract specification to execution
  - End-to-end eScience workflow operation

Our Scientific Contributions 2

- Trial and tested robust workflow machine copes with changes
  - Arbitrary changing software modules at core of operations
  - Arbitrary changes in computational environment
  - Tested at large scale 200-5000 cores
- Fault tolerance and recovery
  - Algorithms for autonomous job error detection, error handling and job recovery working closely with workflow manager
  - Parameter-based heuristics and knowledge based error handling algorithms
  - Error handling seamlessly over a heterogeneous HPC environment

Conclusion

- Quality of Resilience (QoR) framework is highly relevant and of great interest to workflow performance analysis and system quality assurance
- NCHC will host the F4K data and system for further 2 years, convenient for project expansion!!
- Useful work can be done, if funding available !!

Demo Video – An Integrated F4K System Powered by the Workflow Engine

Cheng-Lin Yang
Jessica Chen-Burger
Gaya Nadarajan

University of Edinburgh
EU Fish4Knowledge Project

Thank you for listening!!