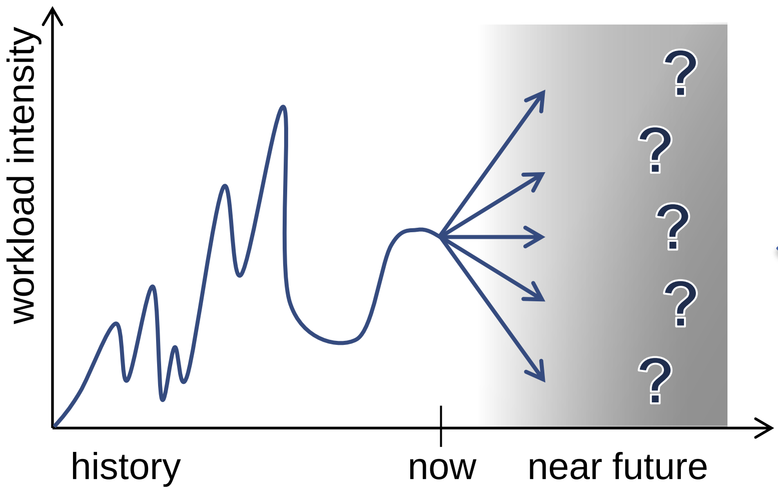
Workload Classification and Forecasting

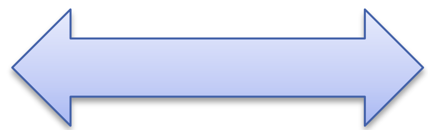
Nikolas Roman Herbst, Nikolaus Huber, Samuel Kounev, Erich Amrehn (IBM R&D)

KoSSE-Symposium on Application Performance Management **November 29th, 2012**

SOFTWARE DESIGN AND QUALITY GROUP
INSTITUTE FOR PROGRAM STRUCTURES AND DATA ORGANIZATION, FACULTY OF INFORMATICS

Motivation





Forecast approaches of the time series analysis:

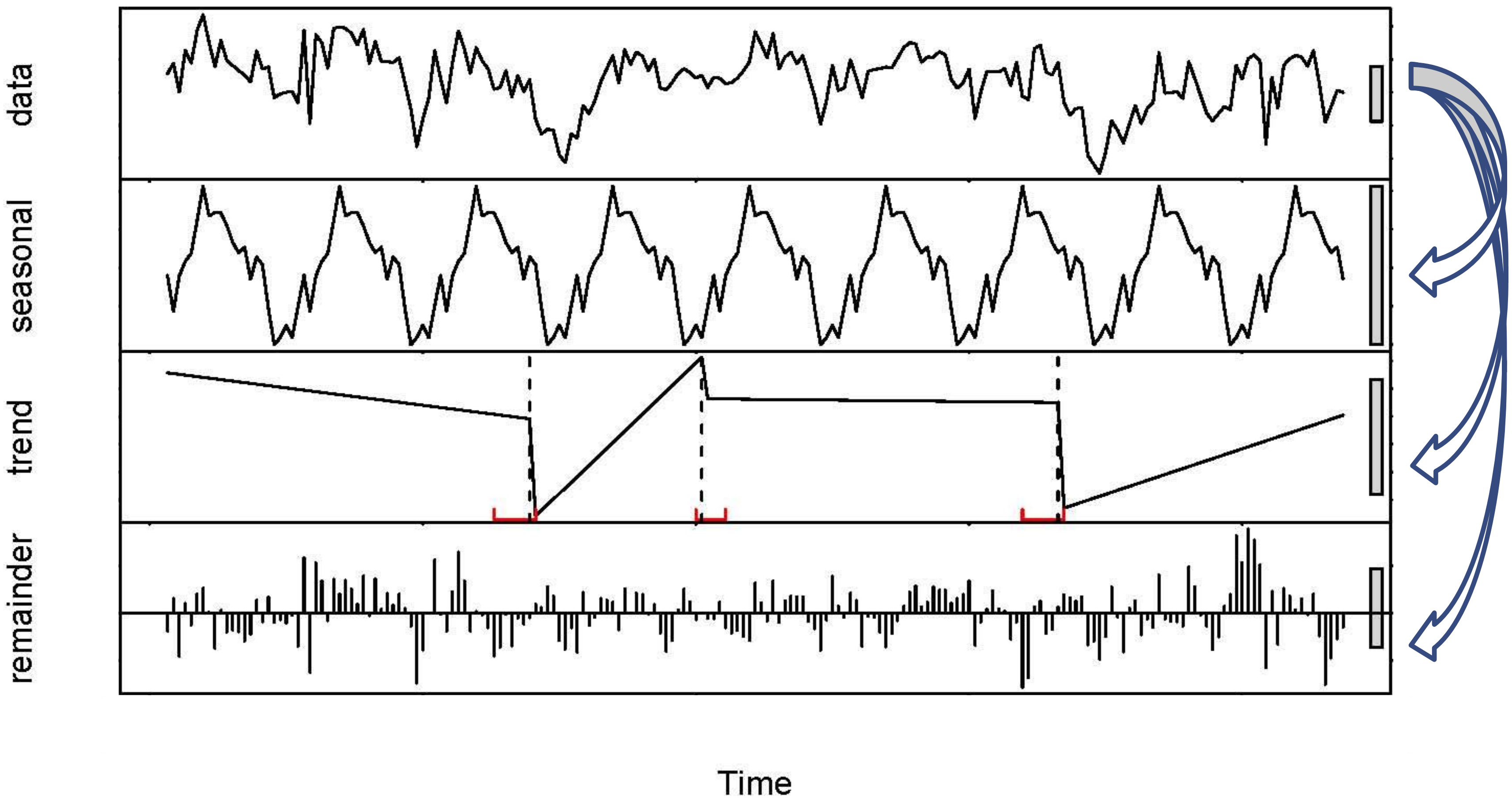
Idea: Intelligent and dynamic use of different tools out of the toolkit

Goal: Providing information on most likely future developments

„Knowing about a problem before feeling it"

1) Which tools are in the toolkit? **properties - requirements - strength**?
2) How can we characterize possible **scenarios**?
3) How do we **select and apply** a tool in a certain scenario?
4) **Direct Feedback**: … Did we select the most appropriate tool and was it beneficial?

Time Series Analysis



[BFAST]

Forecasting Strategies

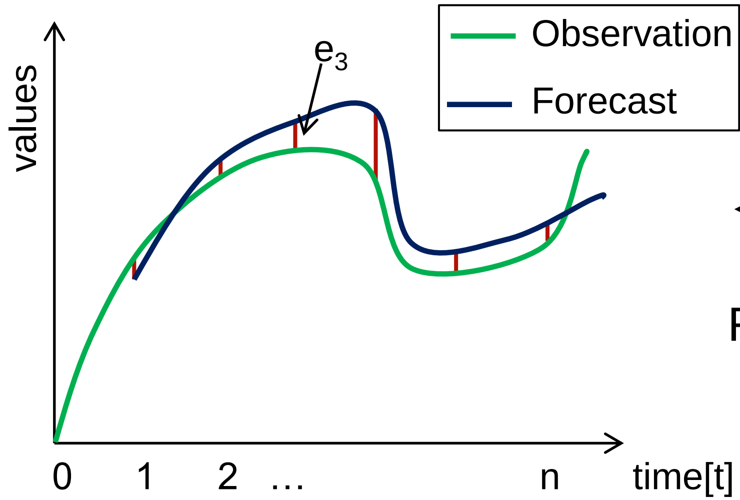
Basic Methods	(initial)
Naïve, Moving Averages, Random Walk	

Trend Interpolation	(fast)
Simple Exponential Smoothing (SES)	[Hynd08]
Cubic Smoothing Splines	[Hynd02]
Croston's method for intermittent time series	[Shen05]
Autoregressive Moving Averages (ARMA11)	[Box08]

Estimation and Modelling of Seasonal Pattern	(complex)
Extended Exponential Smoothing (ETS)	[Hynd08, Hyn08]
ARIMA framework with automatic model selection	[Box08, Hynd08]
tBATS for complex seasonal patterns	[Live11]

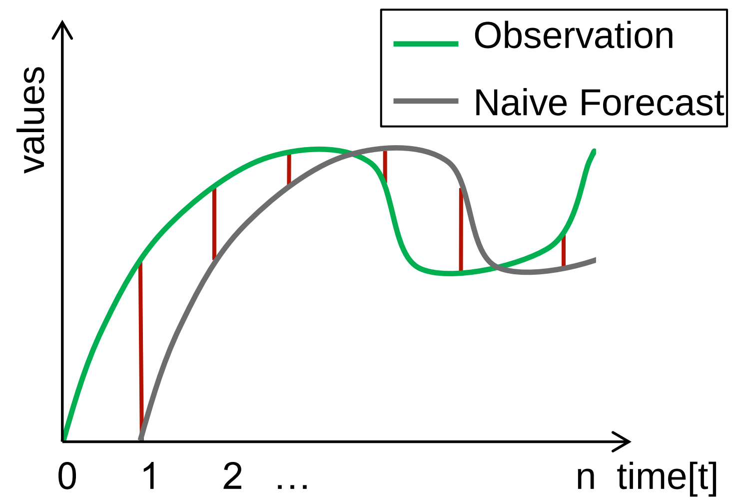
Forecast Accuracy Metric

Mean absolute scaled error (MASE) [Hynd06]



◀ Forecast

Reference ▶



$$e_t = forecastValue_t - observedValue_t$$

$$b_n = \frac{1}{n} \times \sum_{i=1}^{n} |observedValue_i - observedValue_{i-1}|$$

$$mase(0; n) = mean_{t=\{1;n\}}(|\frac{e_t}{b_n}|)$$

Workload Intensity Characterization

Karlsruhe Institute of Technology

High level data analysis to gain information on:

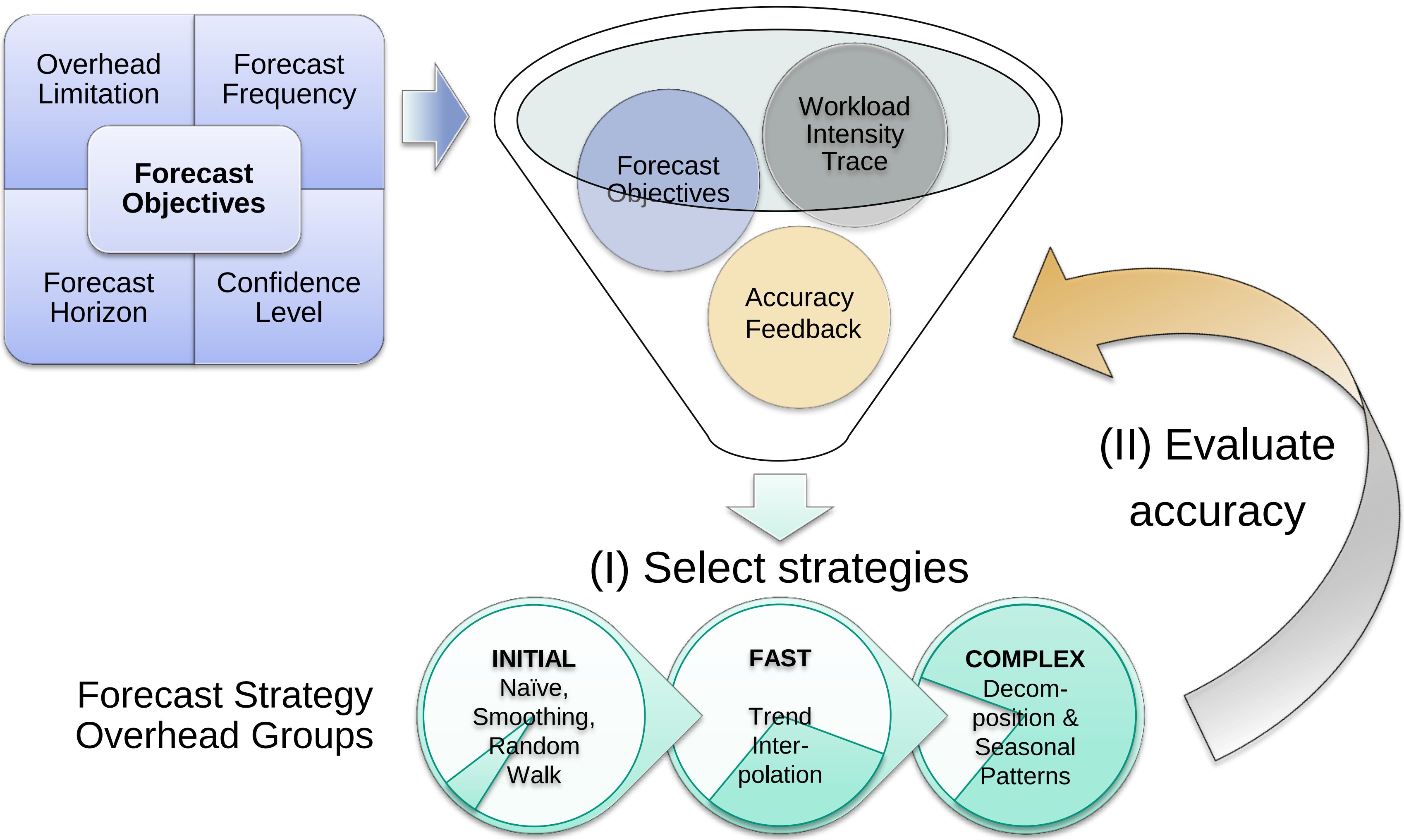
- Noise level & occurrences of unpredictable bursts
- Influence of trends and seasonal patterns

Workload Intensity Trace:
Time Series of Request Arrival Rates

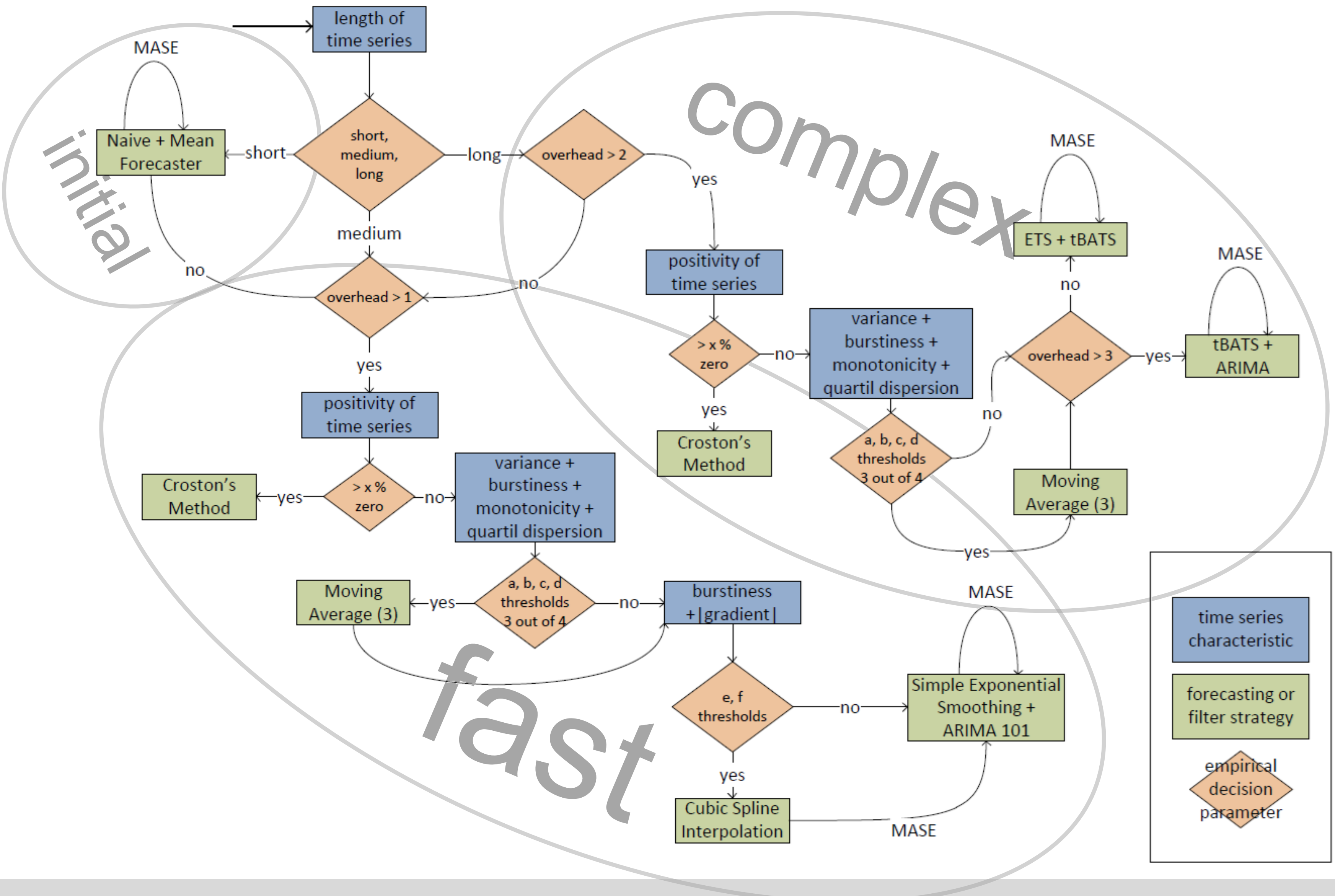
Length | Positivity | Frequency

Gradient | Monotonicity | Variance | Quartile Dispersion | Burstiness

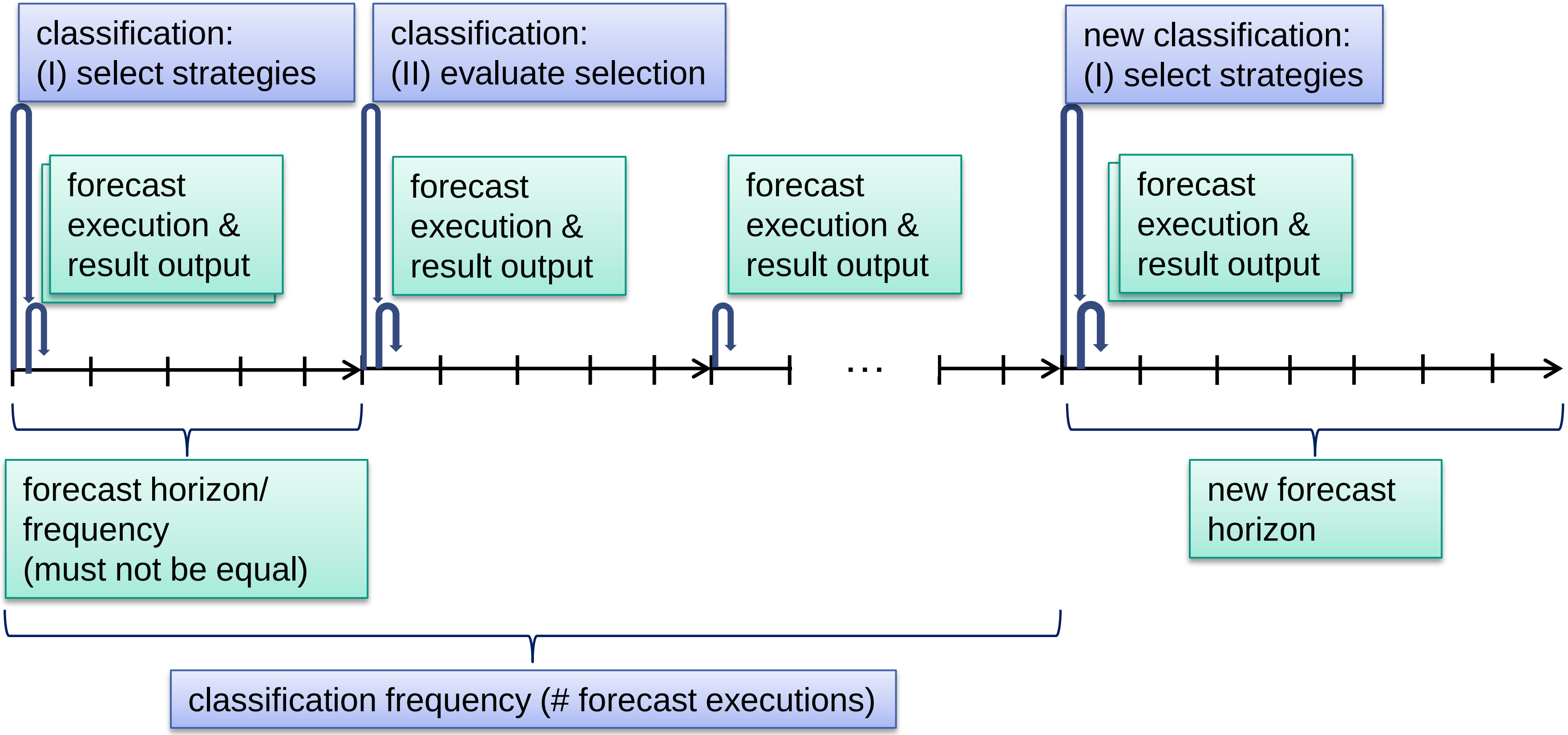
Classification Mechanism



Decision Tree for Classification



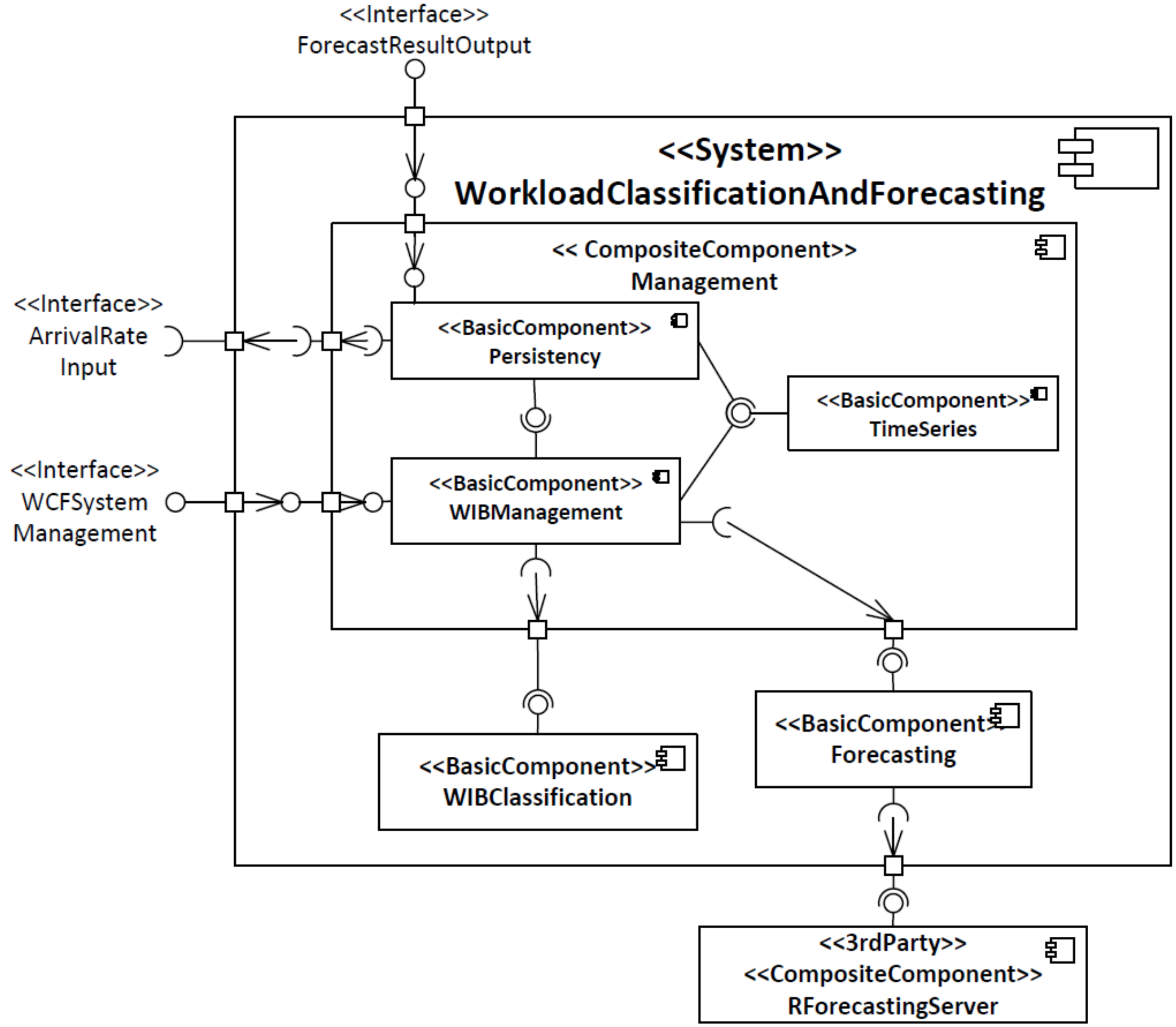
Data, Timing, Parameters



Data input stream:
- Time series of request arrival rates [0; maxSize] most recent values time unit, delta time & start time

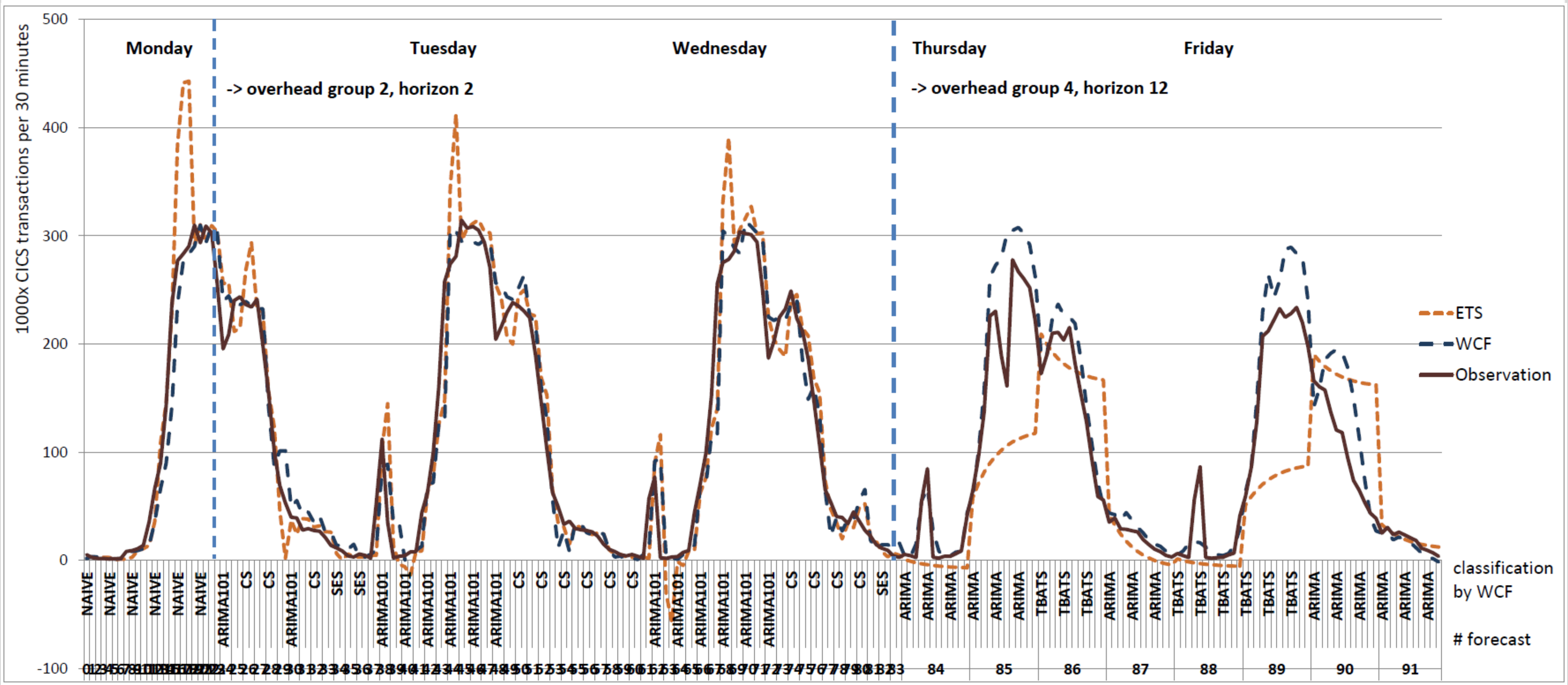
Result output stream:
- Time series of forecast mean values, confidence intervals & MASE metric

Architecture and Implementation



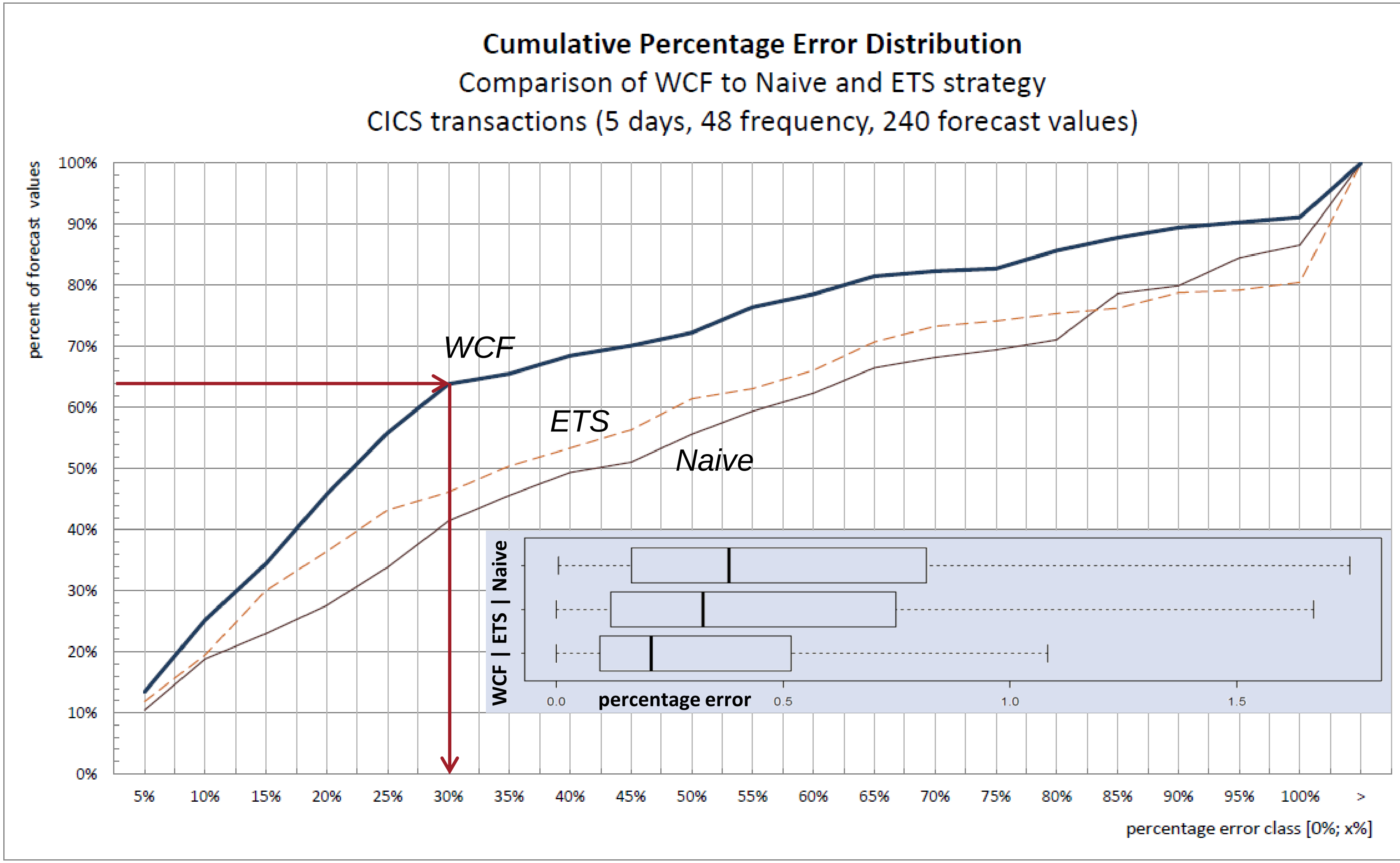
Experiment: Example for Forecast Accuracy Improvement

- Real-world workload intensity trace (IBM CICS transactions on System z)
- Comparison of **Workload Classification & Forecasting (WCF)** approach to **Extended Exponential Smoothing (ETS)** and **Naive** forecast



Software Design and Quality Group
Institute for Program Structures and Data Organization

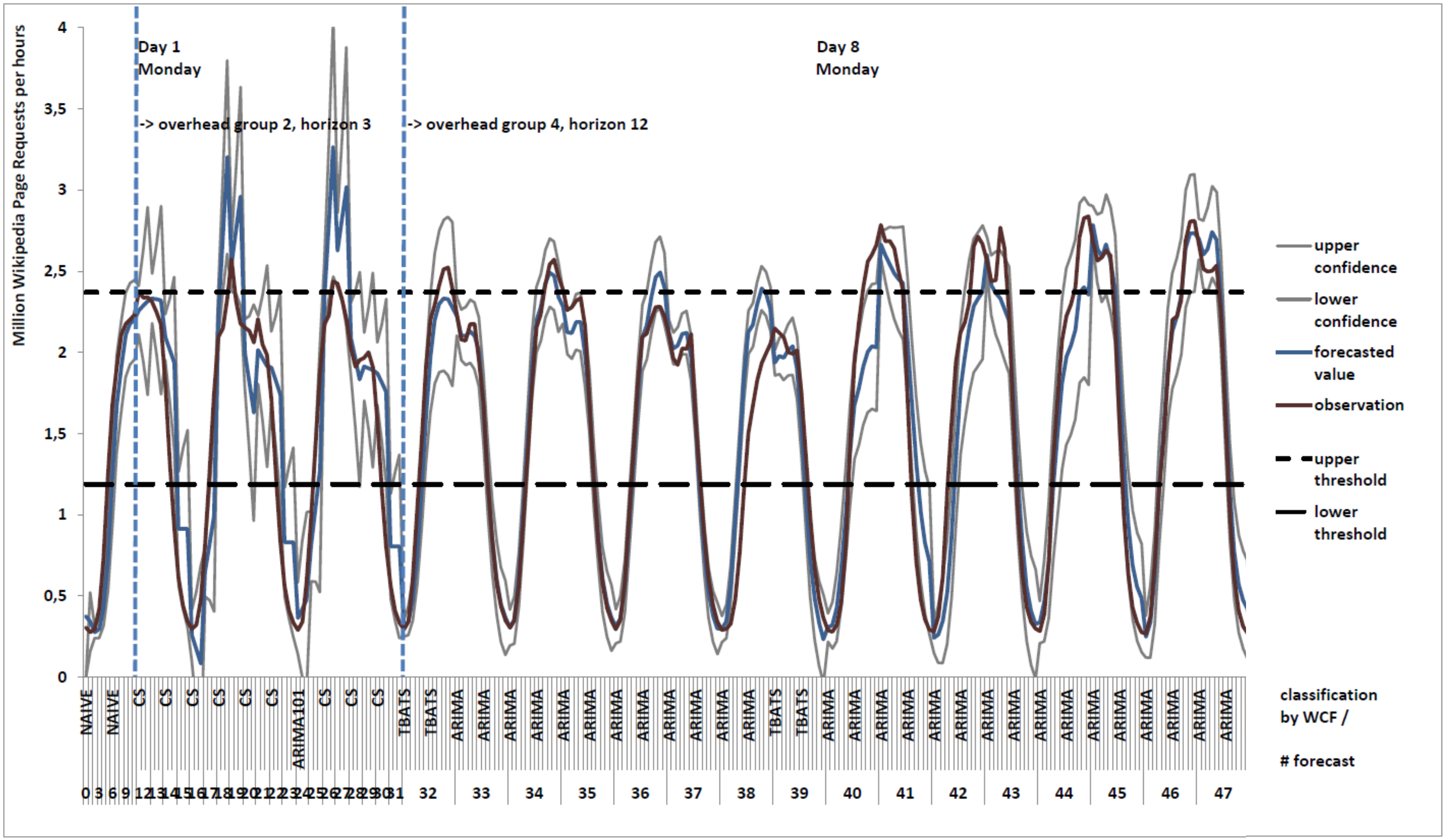
Experiment



Foundations › Approach › Architecture › **Evaluation** › Related Work › Summary

Case Study: Example for Using Forecast Results

- **Scenario:** Additional server instances at certain thresholds, 3 weeks
- Real-world workload intensity trace (**Wikipedia DE** page requests per hour)



Case Study

Resource provisioning:

(I) Without forecasting (solely reactive):

Resource provisioning actions triggered by

76 SLA violations

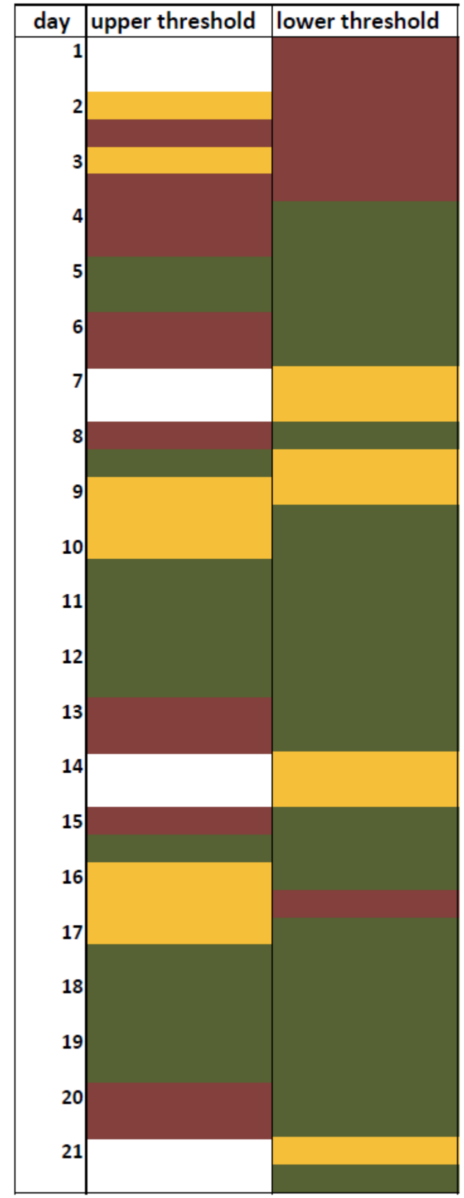
(II) Interpreting WCF forecast results (add. proactive):

Reduction to **34 or less SLA violations**

→ No significant change in resource usage observed

(server instances per hour)

8x	correct forecast:	server instance not needed
42 x	correct forecast:	server instance needed at time t
15 x	nearly correct forecast:	time t slightly too early or too late
19 x	incorrect forecast:	need not detected or false positive



Related Work

Time Series Analysis

[Box08, Hynd08 Shum11]

Workload & Performance Monitoring

e.g. [vHoo12]

Forecasting for Proactive Resource Provisioning

[Grun12, Hedw10, Gmac08, Mena04, Bens07]

Self-Adaptive Resource Provisioning

e.g. [Hube11, Koun10]

→ Focus on forecasting of **performance metrics** not **workload intensity**

→ Focus on **single tools** of the toolkit or other toolkits no **dynamic selection**

Summary & Outlook

Survey on Forecast Approaches

Implementation of the WCF-System
provides continuous forecast results at run-time

Forecast Accuracy Improvement:
\> **37 %** compared to ETS as an established approach

Proactive Resource Provisioning enabled:
\> Up to **75 %** less SLA violations than reactive

Construction of a Workload Classification Scheme

Experiments and Case Study:
Evaluation based on real-world workload intensity traces

Future Work:
\> System Integration with Kieker
\> Filters: Objective Selection, Splitting
\> Use for Anomaly Detection [Biel12]

Literature

[Bens07] M. Bensch, D. Brugger, W. Rosenstiel, M. Bogdan, W. G. Spruth, and P. Baeuerle, Self-learning prediction system for optimization of workload management in a mainframe operating system," in ICEIS 2007

[BFAST] R Package: BFAST, Breaks for additive Season and Trend, http://bfast.r-forge.r-project.org/

[Biel12] T. C. Bielefeld, Online Performance Anomaly Detection for Large-Scale Software Systems, March 2012, Diploma Thesis, University of Kiel

[Box08] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, Time series analysis : forecasting and control, 2008

[Gmac07] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, Workload analysis and demand prediction of enterprise data center applications, IISWC '07

[Grun12] A. Amin, A. Colman, and L. Grunske, An Approach to Forecasting QoS Attributes of Web Services Based on ARIMA and GARCH Models, ICWS 2012

[Hedw10] M. Hedwig, S. Malkowski, C. Bodenstein, and D. Neumann, Towards autonomic cost-aware allocation of cloud resources, ICIS 2010

[Hube11] N. Huber, F. Brosig, and S. Kounev, Model-based Self-Adaptive Resource Allocation in Virtualized Environments," SEAMS 2011

[Hynd02] R. J. Hyndman, M. L. King, I. Pitrun, and B. Billah, Local linear forecasts using cubic smoothing splines, Monash University, Department of Econometrics and Business Statistics, 2002

Literature

[Hynd06] R. J. Hyndman and A. B. Koehler, Another look at measures of forecast accuracy, International Journal of Forecasting

[Hynd08] R. J. Hyndman and Y. Khandakar, Automatic time series forecasting: The forecast package for R 2008

[Hyn08] R. J. Hyndman, Koehler, Forecasting with Exponential Smoothing : The State Space Approach, Springer Series in Statistics, Berlin, 2008

[Koun10] S. Kounev, F. Brosig, N. Huber, and R. Reussner, Towards self-aware performance and resource management in modern service-oriented systems, SCC '10

[Live11] A. M. De Livera, R. J. Hyndman, and R. D. Snyder, Forecasting time series with complex seasonal patterns using exponential smoothing, Journal of the American Statistical Association, vol. 106, no. 496, pp.1513, 2011

[vHoo12] A. van Hoorn, J. Waller, and W. Hasselbring, Kieker: A framework for application performance monitoring and dynamic software analysis," ICPE 2012

[Mena04] M. N. Bennani and D. A. Menasce, Assessing the robustness of self-managing computer systems under highly variable workloads, 2004

[Shen05] L. Shenstone and R. J. Hyndman, Stochastic models underlying Croston's method for intermittent demand forecasting," Journal of Forecasting, vol. 24, no. 6, pp. 389, 2005

[Shum11] R. H. Shumway, Time Series Analysis and Its Applications: With R Examples, Springer

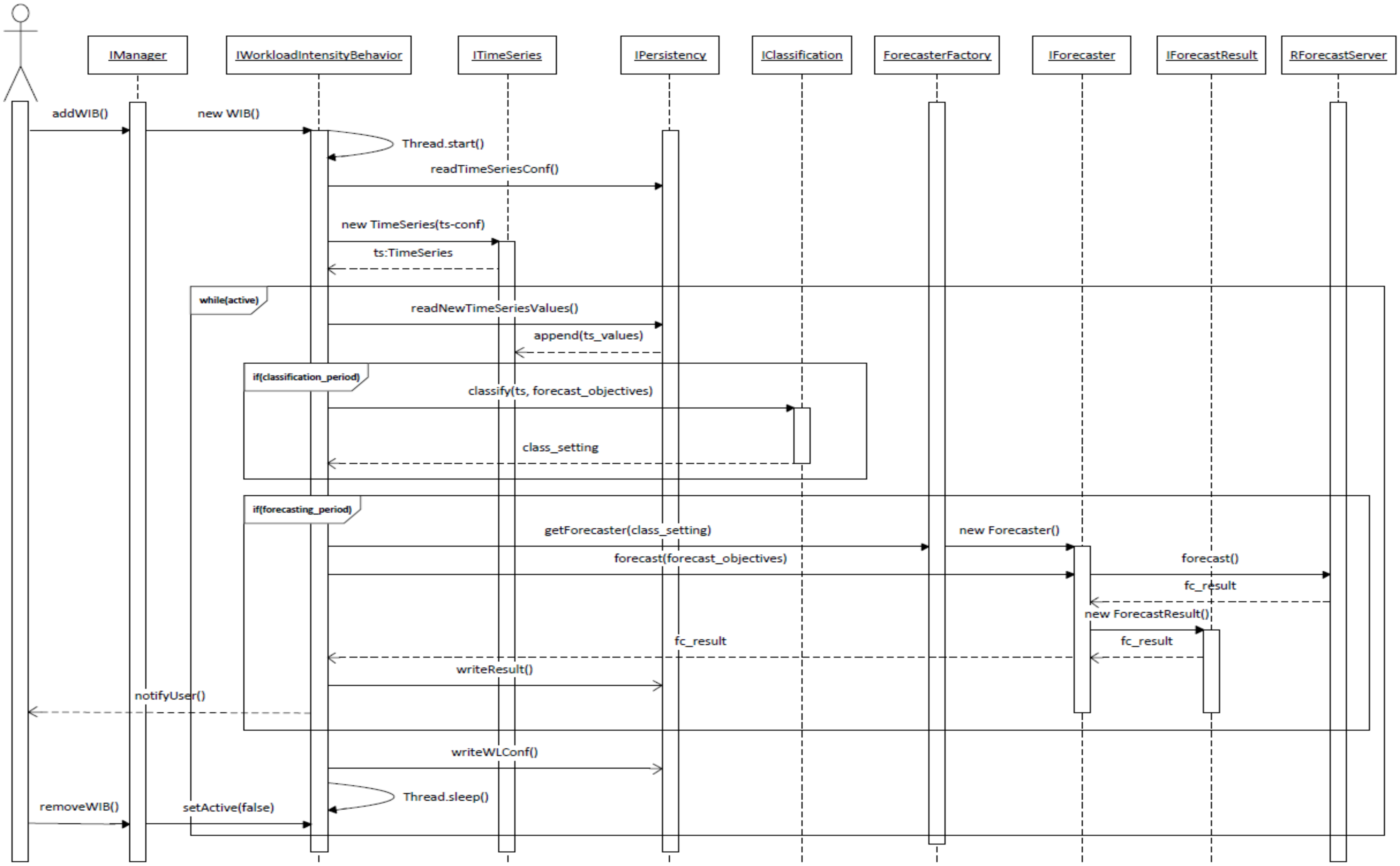
Backup: Forecast Objectives

parameter name	parameter space	proposed setting	explanation
forecast period	[1;max_int]	[1; *frequency*]	This objective defines how often a forecast is executed in times of new time series points. For a value of 1 a forecast is requested every new time series point and can be dynamically increased by period factors in the classification setting to reach the configured maximum horizon. This value should be equal or smaller than the *start horizon* objective (if continuous or even overlapping forecasts are needed)
highest overhead group	[1;4]	[2;4]	This objective defines the highest overhead group from which the forecast strategies will be chosen. A value of 2 may be sufficient if the time series data have strong trend components that are not overlaid by seasonal patterns, as the strength of class 2 strategies is the trend extrapolation. For time series with seasonal patterns, a setting of 3 for a maximum forecast computation time of 30 seconds and 4 for forecast computation times below 1 minute is recommended.
confidence level	[0;100)	may be given by a forecast interpreter	The confidence level α of the returned forecast confidence intervals is defined by this objective.
start horizon	[1;max_int]	[1; 1/8x *frequency*]	The *start horizon* defines the number of time series points to be forecasted at the beginning and can be dynamically increased by period factors in the classification setting up to the *maximum horizon* setting. This value should be equal or higher than the *forecast period* objective (if continuous or even overlapping forecasts are needed).
maximum horizon	[1;max_int]	*frequency*	The value of *maximum horizon* setting defines the maximum number of time series points to be forecasted. A recommendation for this setting is the value of the *frequency* setting of the time series, as a higher horizon setting may lead to broad confidence intervals.

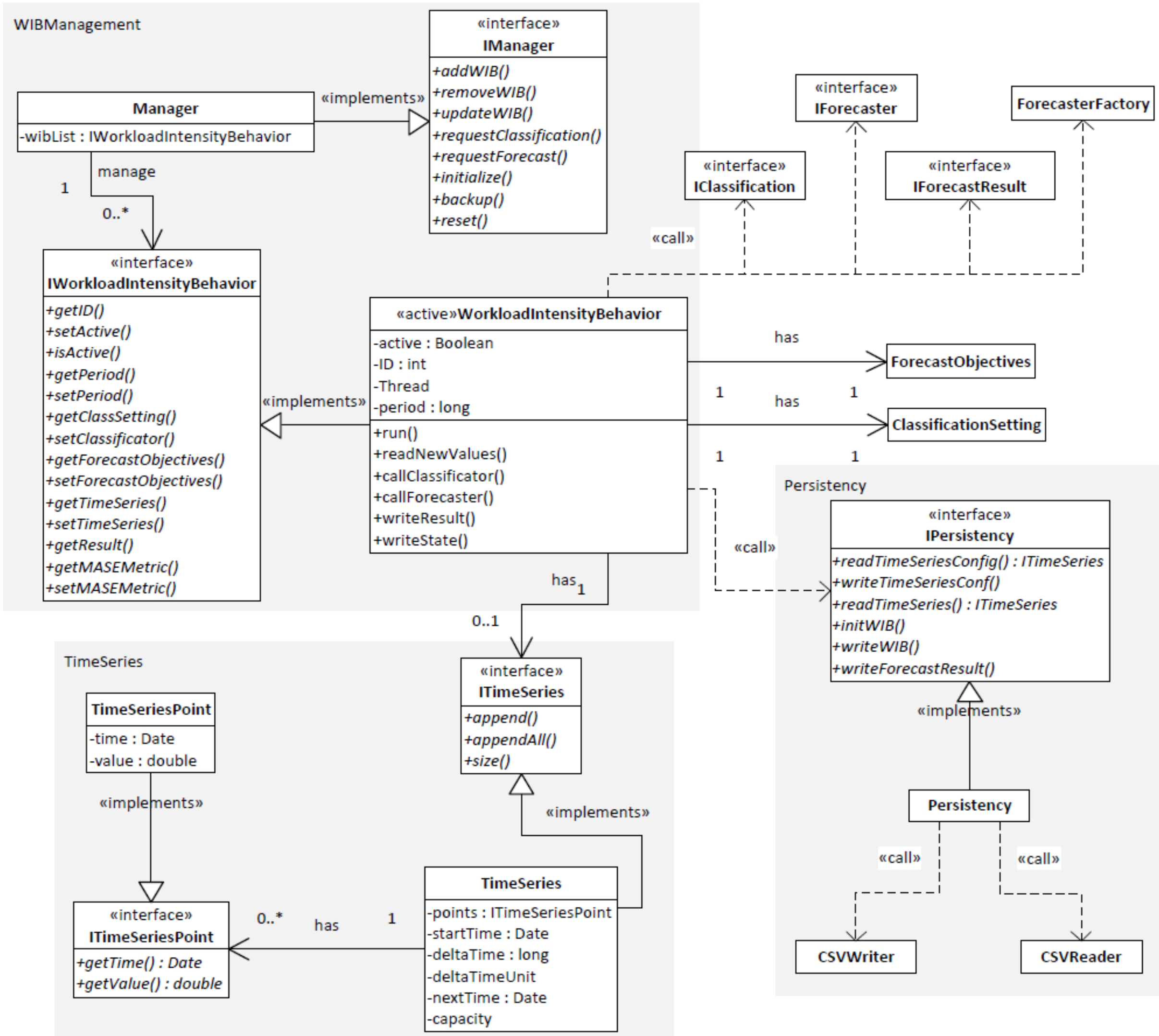
Backup: Forecast Strategy Overhead Groups

overhead group	strategies	application
1 – nearly none	naïve, arithmetic mean	These two strategies are only applied if less than *initial size threshold* values are in the time series. The arithmetic mean strategy can have an forecast accuracy below 1 and therefore be better than a solely reactive approach using implicitly the naïve strategy. This is only true in cases of nearly constant base level of the arrivals rates. These strategies should be executed as frequently as possible every new time series point.
2 - low	cubic spline interpolation, ARIMA 101, simple exponential smoothing, Croston's method for intermitted demands	The strengths of these strategies are the low computational efforts below 100ms and their ability to extrapolate the trend component. They differ in sensitivity to noise level or seasonal components. These strategies need to be executed in a high frequency with small horizons.
3 - medium	extended exponential smoothing, tBATS	The computational effort for both strategies is below 30 sec for a maximum of 200 time series points. They differ in the capabilities of modeling seasonal components.
4 - high	ARIMA, tBATS	The computational effort for the ARIMA approach can reach up to 60 sec for a maximum of 200 time series points and may achieve smaller confidence intervals than the tBATS approach.

Backup: Sequence Diagram

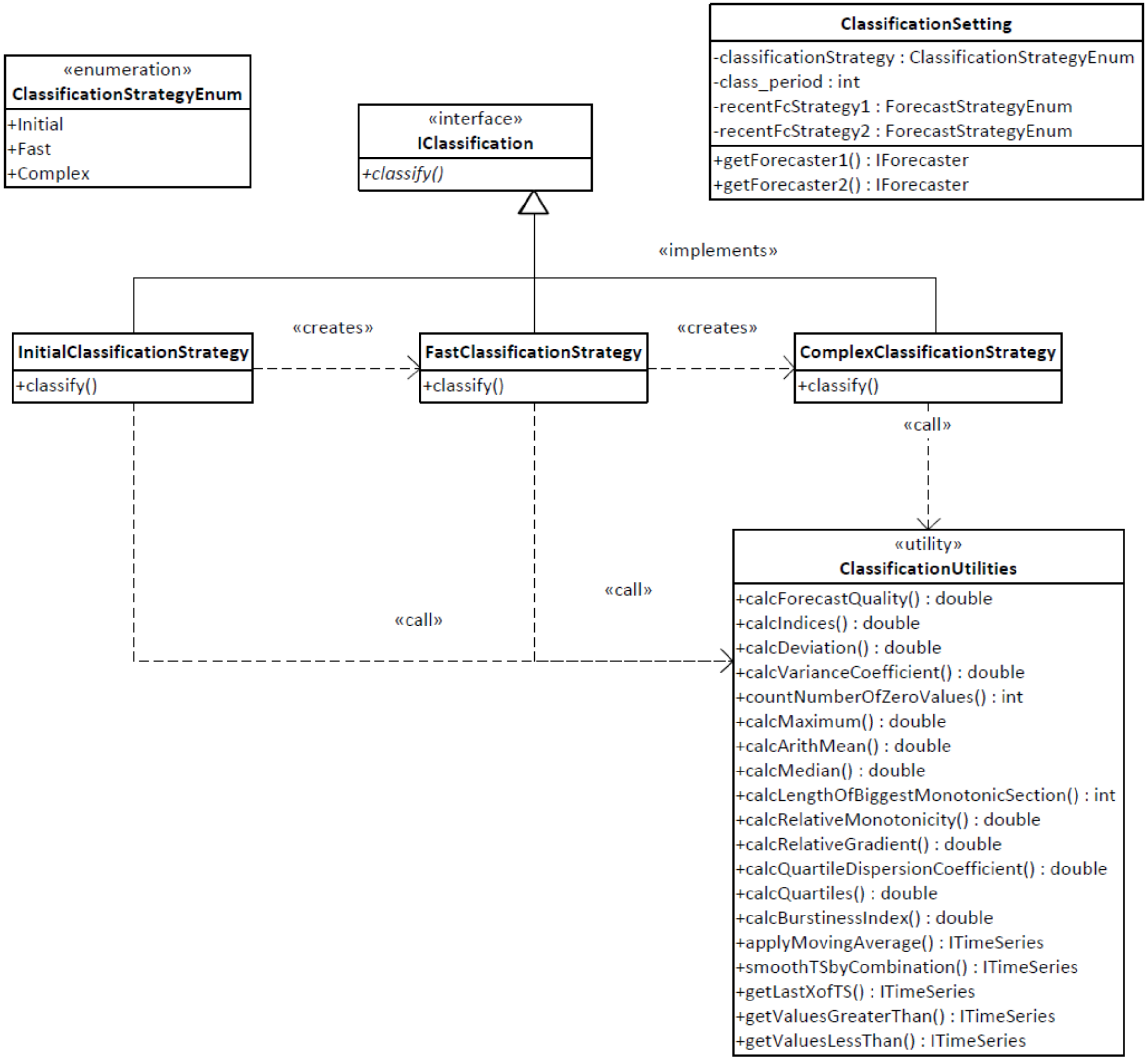


Backup: Class Diagram - Management

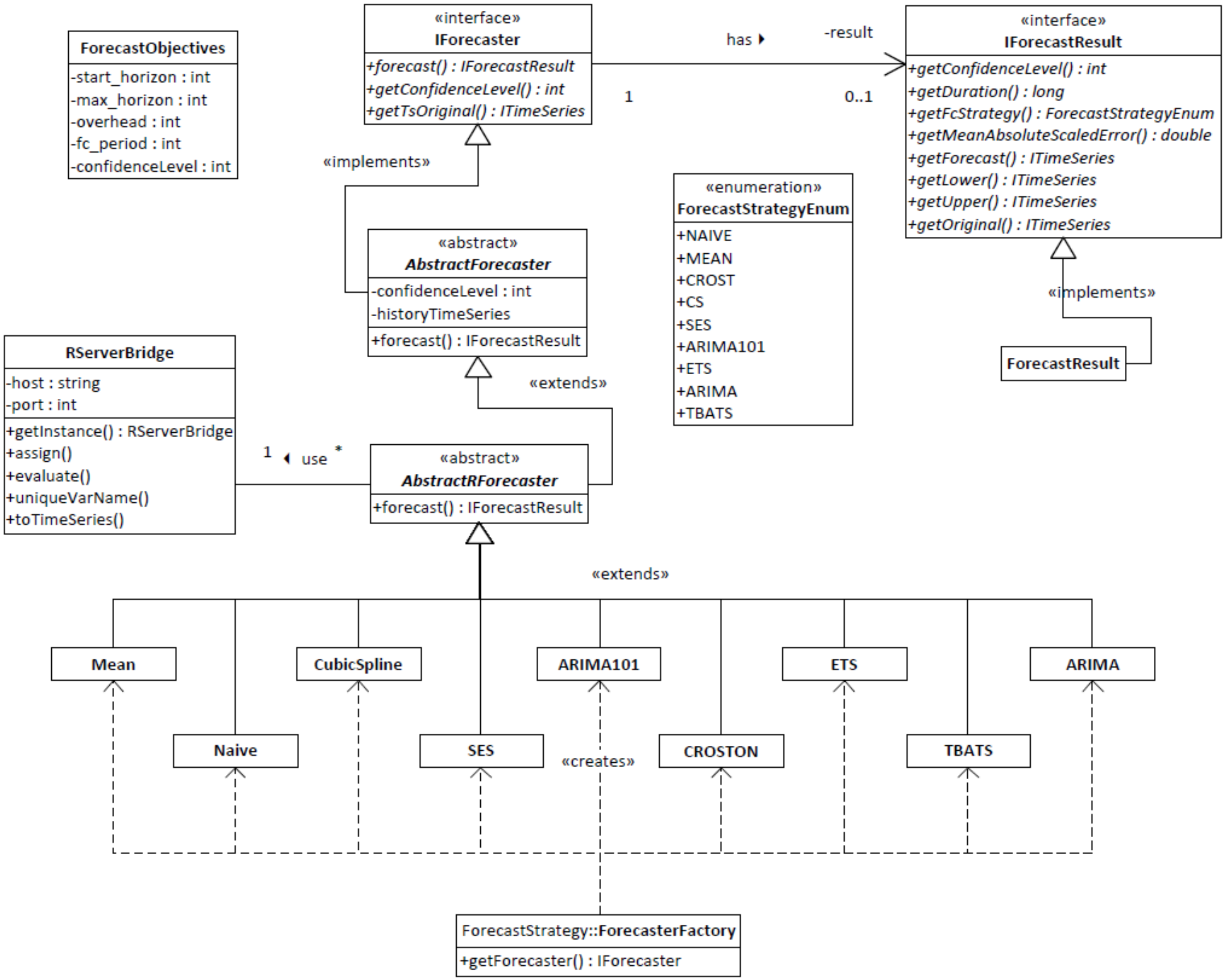


Software Design and Quality Group
Institute for Program Structures and Data Organization

Backup: Class Diagram - Classification



Backup: Class Diagram – Forecasting



Backup: Experiment WCF4 vs. tBATS, ARIMA

WCF limited to choose from tBATS and ARIMA

→ Significant accuracy improvement by combination and dynamic strategy selection