Mindtree
A Larsen & Toubro Group Company

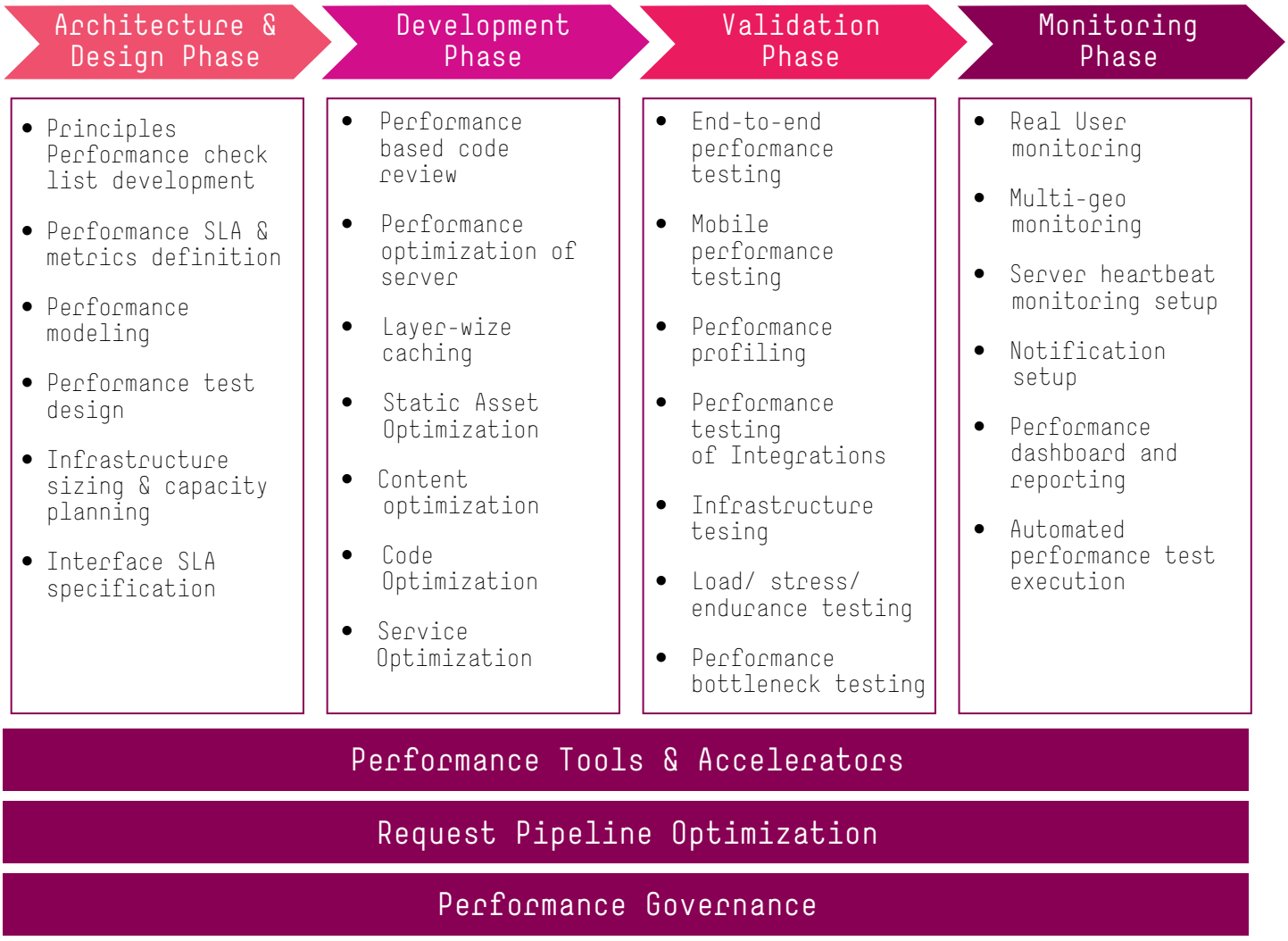
Web Performance Patterns

Introduction

Web performance patterns provide guidance in implementing the best practices based solution. In this white paper, we discuss the performance optimization framework to implement the performance best practices at all the SDLC phases. We also discuss the main performance patterns and anti-patterns.

Welcome to possible

A Mindtree Whitepaper



During the **design and architecture phase,** we could adopt performance-based design. This includes defining performance design principles and developing a performance checklist and performance patterns that could be used during the development and testing phases. We will also finalize the performance SLAs related to response time, throughput, and resource utilization, and such. We would design performance test cases and setup optimal sized infrastructure. Performance modeling and user load modeling of the application are done using peak loads, peak usage hours, application usage patterns, and identifying key performance scenarios and performance objectives and metrics.

During the **development phase,** a performance-based development methodology would be adopted using iterative performance code reviews. Application code, server configurations, and web pages would be fine-tuned from a performance standpoint. The development team would use the performance design checklist and architecture principles defined in the design and architecture phase. The multi-layer caching system would be developed. Asset, service and content optimization techniques would be implemented for page modules.

Performance validation phase involves iterative performance testing and measuring all the identified performance metrics and SLAs for end-to-end performance scenarios. We would conduct various types of performance testing such as peak load testing, infrastructure testing, endurance testing, infrastructure and content volume testing. During the performance testing and analysis phase, we would identify performance bottlenecks and fine-tune components and systems to address the bottleneck. Performance testing will be conducted on all supported browsers and mobile devices.

In **post-production phase,** we would mainly conduct performance monitoring activities such as server health check monitoring, real-time application monitoring, automated performance testing. SLA violations will be reported and notified pro-actively to system administrators to take corrective actions.

Horizontal components such as performance tools and accelerators would be used out in all phases for enhanced productivity. Web performance governance would span all phases of web performance optimization. The web performance would be optimized at various phases of the web request processing pipeline .

Web Performance Patterns and Anti-patterns

Identify the key performance anti-patterns and address those gaps.

Web Page Design Anti Patterns

- **Bad design of key pages** (such as gateway pages, homepages, or landing pages) by including numerous images, entry popups, heavy banners, banner ads, and presentation components.

- **Cluttered and heavy landing pages** that are not targeted for user personas and complex page design without unified interface. Using a huge number of resource requests impacting parallel downloads and having banner ads, entry popups, and too many calls to actions, and clickable links in landing pages.

- **The absence of real-time performance monitoring** and notification infrastructure and absence of layer-wise caching strategy.

- **Using uncompressed images and scripts** on the pages.

- **Bad integration design**

    - 3rd Party component integration without proper SLA framework

    - Improper handling of timeouts and exceptions in the services

- **Having front-end Single Point Of Failure (SPOF)** such as blocking 3rd party script,

synchronous load, delayed/long-running blocking JavaScript, in-lined font-face, in-lined scripts, and in-lined images that prevent browser caching, edge caching, and on-demand loading and increases load time by 16%.

- **Huge white space** in HTML document increases page size.
- **Absence of Omni-Channel strategy**
  - Absence of mobility enabled sites or lack of multi-device testing
  - Absence of cross-browser testing
- **Absence of early and iterative performance testing**
- **Other common issues with page performance are as follows**
  - Numerous JS/CSS includes
  - Duplicate calls
  - Broken links
  - Unnecessary calls
  - Placement of JS/CSS calls
  - Bloated size of web page
  - Frequent resource requests with huge payload
  - Inline styles and JS logic

Page Design Patterns and Best Practices

Given below are the performance best practices and performance design patterns that are applicable in the design phase of the project:

- **Avoiding extraneous content** such as ads can improve the number of objects per page by 25% and improves latency by 30%.
- **Adopting user-centric design approach** addresses challenges related to usability, information discovery, accessibility, task completion.
- **Conducting iterative performance testing** assesses the page performance across geography and Omni-channel testing for all pages.
- **Adopting user-friendly and intuitive information architecture** and minimizing

pages/links needed to find the information or to reach the correct page. Create information architecture and page flows based on user goals and personas so that users can reach the information quickly and complete the intended task.

- **Keep the key pages simple in design.** This involves using only necessary UI components. Complex page design and page cluttering should be avoided. Optimize the landing page through techniques such as eyeball tracking, uncluttering, targeted and useful information, and A-B Split testing/multivariate testing analysis. The right pane elements can load late as its performance is less critical.

- **Use Responsive Web Design (RWD) technique** to cater to multiple devices and form factors. RWD consists of fluid grids, media queries that can auto-adjust based on the target device specifications. Users perceive instantaneous response time (0.1-0.2 seconds), and they feel that that information interaction time is 1-5 seconds; hence it is important to adopt the responsive design to create interactive and highly performing UI elements.

- **Minimize page weight.** Preferably the overall page size should be between 100KB and 400KB for home pages and landing pages. Minimize session size and cookie size.

Business-critical processes should be optimized. This includes business process optimization, page design optimization, search optimization, check out/shopping process optimization, user registration optimization and such.

- **Remove known performance blockers** such as numerous unnecessary links, iframes, numerous pages, and non-intuitive information architecture.

- **AJAX-enable** the web applications to fetch the resources and to load the page data. It results in more responsive and shorter inter-request times and burstier traffic.

Design of Server Calls

Given below are the main anti-patterns that occur during server calls and back-end services invocation:

- Explore ways to load the page content asynchronously. We can leverage AJAX requests to load the page sections, which provide non-blocking page loads.

- Ensure the page data is loaded only on demand and in lazy mode. For instance, the list data or results data can be shown in paginated view and can be loaded only on user navigation.

- Use asynchronous scripts and AJAX gets requests.

- Specify design goals for external and 3rd party scripts. The main design goals for the external

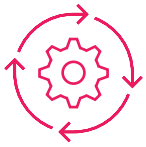
scripts are small size, readable, unobtrusive, and easy to copy-paste to the host page and asynchronous support.

Web Performance Patterns

Given below are some of the key web performance patterns for optimal performance:

- **Make web components lighter,** move them closer to the layer where it is used, cache them longer, and load them more intelligently.
- **Layer-wise caching** at all layers in the request processing pipeline for optimal performance.
- **Progressive enhancement technique** that uses layers of standards such as XHTML, CSS JavaScript to overlay dynamic content with CSS, JavaScript to provide cross-browser accessible content. The technique mainly consists of the behavior layer (implemented through unobtrusive JavaScripts), presentation layer (implemented through CSS), structure layer (implemented through HTML standards) and core content layer and these layers are selectively added based on the device capability to maximize usability and accessibility.
- **Minimal round trips:** The web page should minimize the server calls to the extent possible. Wherever possible, the calls should be batched to minimize the calls.
- **Asynchronous loading pattern:** All the page assets should be loaded asynchronously and resource requests should adopt asynchronous communication.
- **Lazy loading pattern:** The page assets should be loaded when required and on-demand.
- **Lightweight design:** The page should adopt a lean model using web-oriented architecture and use light-weight integration technique.
- **Device specific rendition:** The page content, assets should be optimized for the rendition device.
- **Responsive page content:** Responsive design for HTML elements and adaptive design for content should be followed.

Web Architecture patterns



Model-View-Controller architecture style

Microservices Architecture

Web-oriented Architecture

The model-view-controller architecture style is the widely used architecture pattern that creates loosely coupled flexible web applications with modular components. From the performance viewpoint, interactional styles such as event observation and notification, publish/subscribe and asynchronous communication can be added as features for MVC applications. Most modern web applications heavily use the Representational State Transfer (REST) architecture style which provides light-weight and asynchronous methods for requesting and updating web resources.

Microservices architecture allows us to build a web application as a composition of multiple independently scalable services. The architecture uses light-weight communication mechanism and functional model for building services. Since each of the microservices are individually scalable, we could build a highly scalable and performing system using microservices.

Web-Oriented Architecture (WOA) involves light-weight pluggable client-side widgets. WOA architecture is light-weight in design and we can easily implement the web performance best practices. AJAX-based client-side MVC and Model–View–View Model (MVVM) architectures are used to build rich, interactive and responsive web applications.

About the Author

Dr. Shailesh Kumar Shivakumar
Solution Architect

Dr. Shailesh Kumar Shivakumar has 19+ years of experience in a wide spectrum of digital technologies including, enterprise portals, content management systems, lean portals, and microservices. He holds a Ph.D. degree in computer science and has authored eight technical books published by the world's top academic publishers such as Elsevier Science, Taylor and Francis, Wiley/IEEE Press, and Apress. Dr. Shailesh has authored more than 14 technical white papers, five blogs, twelve textbook chapters for various undergraduate and post-graduate programs and has contributed multiple articles. He has published 20+ research papers in reputed international journals. Dr. Shailesh holds two granted US patents, apart from ten patent applications. Dr. Shailesh has presented multiple research papers at international conferences.
Dr. Shailesh's Google Knowledge Graph can be accessed at **https://g.co/kgs/4YoaiN .** He has successfully led several large scale digital engagements for Fortune 500 clients. Shailesh can be reached at **Shaileshkumar.Shivakumarasetty@mindtree.com**

About Mindtree

Mindtree [NSE: MINDTREE] is a global technology consulting and services company that enables enterprises across industries to drive superior competitive advantage, customer experiences and business outcomes by harnessing digital and cloud technologies. A digital transformation partner to more than 260 of the world's most pioneering enterprises, Mindtree brings extensive domain, technology and consulting expertise to help reimagine business models, accelerate innovation and maximize growth. As a socially and environmentally responsible business, Mindtree is focused on growth as well as sustainability in building long-term stakeholder value. Powered by more than 29,700 talented and entrepreneurial professionals across 24 countries, Mindtree — a Larsen & Toubro Group company — is consistently recognized among the best places to work.
For more, please visit **www.mindtree.com** or **@Mindtree_Ltd.**