Microservices & Containers
A Pneuron Perspective
# Table of Contents

Context ........................................................................................................................................... 3
The Rise of Microservices .................................................................................................................. 5
Architectural Concept of “Classic” Microservices ........................................................................... 6
What Makes “Classic” Microservices a Significant Commitment? ................................................ 7
The Pneuron Platform - "Microservices & Containers for the Rest of Us" .................................... 8
Pneuron - A Highly Effective Entry Point for Microservices-Style Solutions ............................ 12
References ......................................................................................................................................... 13
About Pneuron ................................................................................................................................. 13
Context

Never have the demands on business been greater. Global markets, non-traditional competitors, rapidly escalating regulatory regimes, and the introduction of new technology are but a few of the many significant challenges that organizations are facing in the 21st century. While past history has shown that the most successful businesses have a great ability to adapt to change, a strong argument could be made that the breadth, depth, and pace of change across all of these dimensions is far higher and more impactful now than in past generations - making it harder to keep up than ever before.

From a business perspective, firms have been experimenting with dozens of innovative management practices to stay competitive. Outsourcing, offshoring, and e-Commerce are just a few of the significant strategic changes being implemented by management. Importantly, new technology is frequently called upon to either solve existing challenges or create new opportunities in the face of these changes.

Technology in the C-Suite Today

CFO

57% of CFOs agree that predictive analytics creates a key competitive advantage, but 84% say it is difficult to control the quality of their financial data.6

CEO

CEOs of outperforming enterprises are 33% more likely than their peers to make emerging technology their top strategic priority over the next three years.1

CIO

84% of CIOs are focusing investment on new technologies that derive insight and intelligence to connect with customers.3

CMO

Gartner predicts that by 2017, CMOs will spend more on IT than CIOs.4

Visit our website at www.pneuron.com or email us at pneuron@ust-global.com
Regardless of a firm’s use of technology, there are a set of business imperatives that serve as a critical foundation for success:

- **The speed** with which business can create and deliver new forms of value.
- **The agility** of how a business adapts to change.
- **The efficiency** with which a business converts inputs to outputs.

With the influx of highly disruptive technology into nearly all markets, the need for a business architecture that can successfully navigate these challenges is critical. New concepts like “Digital Business” represent a holistic view of how technology is used in all facets of business and is being applied to ensure that the adoption of new technologies supports those three fundamental imperatives.

Critically, the push for a “Digital Business” requires robust alignment between business leaders and technologists as they realize their individual strategic choices are now intimately linked in determining the future success of the firm. Never before has it been so important for business leaders to learn about technology and technology leaders to learn about business.

From a technology perspective, there are similar critical imperatives to ensure readiness to meet today’s business needs. These include:

- **Deployment flexibility** such that new solutions can be executed on the most advantaged platform or service.
- **Adaptability** such that new technology, new requirements, and new sources of value can be readily integrated into existing or new solutions.
- **Dynamic scalability** such that varying workloads can be automatically and cost effectively serviced.

These imperatives provide strategic guidance for the adoption of both new technology and practices that ensure support for the critical needs of the business.

In response to these technology imperatives, a new architectural style for application development has recently emerged: **microservices**. Highly influential

---

A survey of CEOs by PwC revealed the following elements as the “most important ingredients to successful innovation”:

- **57%** The right culture to foster and support innovation
- **44%** Strong visionary business leadership
- **37%** Willingness to challenge norms and take risks
- **31%** Ability to capture ideas throughout the organization
- **31%** Capacity and capability for creativity

According to a recent survey by Bain, companies that perform well in innovation initiatives grow significantly faster than lesser-performing companies with up to a 3x growth difference over 5 years.  

Visit our website at [www.pneuron.com](http://www.pneuron.com) or email us at [pneuron@ust-global.com](mailto:pneuron@ust-global.com)
The evolution of microservices has been driven by the need for unprecedented speed, scalability, and manageability as businesses are reaching activity levels previously unimaginable. Most importantly, both the business and technology characteristics of microservices solutions are highly aligned to the respective imperatives outlined earlier.

Microservices evolved from at least three important technology trends:

- **Prior generations of Service-Oriented Architecture (SOA)** where smaller components of business logic are built as individual services orchestrated by a composite application.
- **Cloud-based deployment** where rapid and highly cost-effective scaling is accessible to virtually any organization.
- **The rise of Software-as-a-Service** which makes access to best-of-breed functionality both highly cost effective and fast to deploy.

Over the last several years, the widespread availability of these technologies has enabled a classic evolutionary path that extracted the best features while leaving behind others that are less suited to new realities. Today, a common set of key tenets exists for microservices-styled architectures that, while not formally encoded as a standard, are accepted as core to this style of solution design. Specifically, most leading practitioners agree on the following:

Visit our website at www.pneuron.com or email us at pneuron@ust-global.com
A focus on the development of **multiple, discrete, “small” elements of functionality** - which simplifies the understanding, development, and testing of logic compared to entire monolithic solutions.

**Individually built, deployed, and scaled services** - which enhances deployment flexibility and supports targeted scaling to optimally address highly varying workloads.

The freedom for each service to **use any appropriate programming language** which allows the use of best fit technology for the required functionality.

The use of **“Containers” as a lightweight solution for packaging developed services** to allow their flexible deployment across diverse computing infrastructures. These containers provide much the same functionality of traditional virtual machines but with significantly less overhead which optimizes the use of compute resources.

The use of **lightweight messaging protocols for communication between discrete services and applications** - which significantly reduces the overhead associated with past use of ESBs and complex integration logic.

Development **teams organized around the full lifecycle of a service** (design, build, run, support) - which supports tighter management and shorter cycles of enhancements or fixes.

**Architectural Concept of “Classic” Microservices**

The following graphic depicts a typical microservices-styled application with centralized **orchestration**. A composite application orchestrates service execution via a series of “calls” utilizing a lightweight messaging protocol. These discretely deployed services can therefore be individually scaled (e.g. Service S3) and each maintains a rigorous functional boundary.

In this classic microservices approach, an application is defined as a centralized body of logic making point to point services calls - in essence, **distributed services with centralized control**.
While a "classic" microservices style of architecture clearly has attractive features from a technology point of view, it carries a number of significant commitments that may well place it beyond the reach of most potential adopters.

As noted above, early adopters are aggressively (and successfully) deploying microservices, but they also have the strategic imperative and commensurate expertise, staffing, and budgets to fully embrace the necessary change – whereas most organizations looking for ways to improve their development capabilities must do so without all the resources normally required.

Let’s look at a few of the most challenging barriers to adopting a full microservices approach, and the impact such change requires:

**Organization of Teams Around Products, Not Projects**
Early adopters are completely re-organizing their development and test resources into smaller, holistically staffed teams that manage the full lifecycle of a given service.

**Migration Toward Advanced Continuous Delivery Principles**
To fully realize those shorter development cycles, firms are greatly advancing their maturity toward continuous development. This requires additional training, tooling, and management skills.

**Services Built for (Only) Specific Solutions and with Limited Centralized Management**
“Classic” microservices are often built to serve only a specific component solution, with re-use relegated to only a secondary priority.

**Introduction of New Container Technologies**
Allowing implementation and runtime support in multiple diverse technologies and platform could bloat the acquisition, infrastructure, and support requirements for core IT operations.

**Investment in Tooling to Manage, Scale, and Handle Disruptions to Services**
Additional or enhanced tooling is required to provide services management, error-handling, and scaling support across the universe of deployed services.
At Pneuron, our founding philosophy was grounded in the same business ambitions advocated by practitioners of what is now called microservices. Specifically, we built a solution development and execution platform to:

- Support **rapid build and deployment cycles** that dramatically shorten time to value
- Offer **containerized functionality** which dramatically simplifies (or even eliminates) integration of diverse enabling platforms
- Provide for **discretely scalable services** which automatically respond to highly varying workloads
- Enable **rapid and highly cost-effective adaptation** to new business, competitive, regulatory, and technology changes
- Significantly **simplify the messaging requirements** across a fabric of functionality and concentrate value-added processing within the various services
- Re-energize the **collaboration between business and IT** in the pursuit of solutions that drive competitiveness

While staying true to many of the “classic” microservices tenants, Pneuron went significantly further. We foresaw that adoption of the techniques now called microservices would require extensive investment in new technologies, re-training of staff, and wholesale changes in management processes.

From the ground up, Pneuron pre-built key supporting technologies and capabilities into our container and across the entire platform to free solution designers, architects, and systems managers from the crippling dependency on integration with other platforms and highly disruptive organizational and process changes. Instead, the Pneuron platform natively handles key operational and management requirements in a consistent and centralized way without inhibiting the creativity and speed promised by the discrete build, deploy, and operation of services.

**The Pneuron Platform - “Microservices & Containers for the Rest of Us”**

With these factors considered, many organizations are hesitating or delaying until microservices and the associated ecosystem of providers and practices matures in the market. While prudent for most, this inevitably delays the opportunities to benefit from advanced practices that, if applied judiciously, could materially improve the price / performance of development organizations in service to their businesses.

Visit our website at [www.pneuron.com](http://www.pneuron.com) or email us at [pneuron@ust-global.com](mailto:pneuron@ust-global.com)
The diagram below shows the architecture of a Pneuron-based solution, with distributed services, execution, and control.

In this approach, an application is defined and deployed as a distributed set of services connected by direct transmission of processing results across an asynchronous execution fabric.

Let’s take a look at just how the components and capabilities of the Pneuron platform build upon or differ from the “classic” microservices architecture.

The Pneuron Difference
In contrast to “classic” microservices, Pneuron solutions are meant to be available for reuse by other solutions. With granular permissioning built into the platform, administrators can easily authorize other designers to use any previously built “services”.

Pneuron offers an extensive library of over 50 (and growing) different types of “Pneurons”. Examples include making a “Query” against a relational database; performing “Analytics” on an incoming set of data or making a webservice call to an application.
The Pneuron Difference

In Pneuron’s case, our container provides a superset of the functionality found in most / all conventional market offerings. As a core platform component (we call it the “Cortex”), it provides a consistent execution environment for all Pneuron-based services (i.e. “Pneurons” and full solutions). Having a single, Java-based container also provides significant simplification across the lifecycle of deployment, management and support, and allows reuse of Java skills commonly found in today’s enterprise. With deployment into a standard JVM, the Cortex provides advanced functionality for resiliency, management, and scalability (discussed below) which eliminates the need for additive platform investments typically required to support a microservices-based architecture.

Containers

<table>
<thead>
<tr>
<th>“Classic” Microservices</th>
<th>Pneuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service designers have the freedom to build and house their services in any type of container.</td>
<td>The container is provided as a core platform component (we call it the “Cortex”) and it provides a consistent execution environment for all Pneuron-based services.</td>
</tr>
<tr>
<td>Provides tremendous freedom to select the most optimal technology for each type of service, although the enterprise must be practical about the support burden incurred from allowing a wide range of container technologies.</td>
<td>The Cortex’s single, Java-based container provides significant simplification across the lifecycle of deployment, management and support, and allows reuse of Java skills commonly found in today’s enterprise.</td>
</tr>
</tbody>
</table>

The Pneuron Difference

Within Pneuron, the execution approach is highly analogous to the “classic” microservices lightweight concept. From a solution logic point of view, the Cortex only understands the configured solution topology while providing various capabilities that support Pneuron execution.

Messaging

<table>
<thead>
<tr>
<th>“Classic” Microservices</th>
<th>Pneuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Composite applications use lightweight messaging protocols to interact with the various services required to meet solution needs.</td>
<td>As execution initiates and continues through successive Pneurons, the Cortex is responsible for the delivery of any Pneuron’s outputs to the one or more destination Pneurons within the solution design.</td>
</tr>
<tr>
<td>Direct, lightweight approach overcomes past challenges of deploying complex and costly integration infrastructure in the form of an Enterprise Service Bus (ESB).</td>
<td>Transmission is done either via a webservices call or a JMS messaging infrastructure - but contains no transformation or supplementary processing (all solution logic is contained within the various configured Pneurons).</td>
</tr>
</tbody>
</table>

The Pneuron Difference

Microservices-based solutions embrace the notion of “dumb pipes and intelligent endpoints” - with individual services performing the value-added processing, while the messaging interface simply provides a pathway for inputs and outputs.
**Execution Model**

<table>
<thead>
<tr>
<th>“Classic” Microservices</th>
<th>Pneuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Composite applications utilize a request / response paradigm to orchestrate required service execution.</td>
<td>There is no central point of orchestration governing service execution. As a Pneuron solution executes, the results of one Pneuron are passed directly to the one or more subsequent Pneurons in the design.</td>
</tr>
<tr>
<td>Each service is tasked with a given set of inputs and responds with the results of the given service logic.</td>
<td>The outputs represent a stateless unit of work, encoded as a self-describing XML message that is delivered asynchronously to any successor Pneuron(s).</td>
</tr>
<tr>
<td>Anticipates and handles the uncertainties across the multiple factors impacting end-to-end performance.</td>
<td></td>
</tr>
</tbody>
</table>

**The Pneuron Difference**

Pneuron's significantly different distributed execution and control model provides tremendous deployment flexibility and architectural freedom for both the solution designer and systems administrator. The Cortex container is typically deployed multiple times in a customer environment. At design time, designers can re-position any Pneuron to run within any installed Cortex providing for a fully customized processing topology aligned to systems proximity, compute resources, or jurisdictional boundaries.

**Resiliency**

<table>
<thead>
<tr>
<th>“Classic” Microservices</th>
<th>Pneuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Designed for failure” to ensure that missing or inoperable services are gracefully handled and do not crash the entire solution.</td>
<td>Solution designers and administrators have the ability to create clusters of Cortices that support high availability objectives, with native functionality that will persist, dispatch, and retire units of work only upon completion of the prescribed logic.</td>
</tr>
<tr>
<td>Additional logic must be built into the solution that will queue, persist, or otherwise ensure work is not lost while waiting for restoration of required services.</td>
<td>In the event of a lost Cortex, the cluster will dynamically reform and interrupted work will be re-queued for execution on that re-formed cluster.</td>
</tr>
<tr>
<td>Can be burdensome to code or may require third party platforms to provide the required functionality.</td>
<td></td>
</tr>
</tbody>
</table>

**The Pneuron Difference**

In addition to clustering, Pneuron also includes the ability to design flexible and customized error handling routines. This error-handling functionality is another included feature in the Pneuron container. Within an executing network, if Pneuron encounters an unavailable system or issue preventing successful Pneuron execution, an error state is triggered and the in-process unit of work will be re-directed to either a specifically designed or generic error-handling network. This handler is itself built as a series of Pneurons that applies any set of functionalities to persist, alert, extinguish or otherwise manage that error.
Many organizations are now considering a microservices style of application architecture to address the modern business needs for speed, agility, and scalability. These imperatives are critical to sustaining competitiveness in modern business, yet full adoption of microservices can be costly and highly disruptive to ongoing operations.

With its emphasis on simplifying the experience of building, deploying, and executing services-based solutions, Pneuron is an excellent entry point for organizations wishing to rapidly innovate, deploy, and scale this new architectural style. By pre-integrating Pneuron - A Highly Effective Entry Point for Microservices-Style Solutions

Learn how you can leverage Pneuron’s advanced containers to rapidly deploy microservices-style solutions - and redesign your business competitiveness.

The Pneuron Difference
Implementing service management capability in a “classic” microservices approach can itself be a major undertaking, and potentially a major barrier for early adopters. But with Pneuron, as each unit of work arrives at the cluster, the dispatcher extracts the intended network and assigned priority and compares those values to the configured cluster. After all invalid “nodes” are filtered out, the dispatcher will execute a configurable routing algorithm (e.g. Shortest Queue, Shortest Predicted Time to Completion) to determine and dispatch that unit of work to the final “best” execution node.

In addition to optimizing the dispatch of work across existing cluster nodes, Pneuron has added dynamic resource scaling to further support high volume processing. With its “realm management” capability, administrators can pre-identify additive capacity (often drawn from a cloud or grid infrastructure) that will be dynamically instantiated and configured into the original static cluster. This additional capacity is triggered when a configurable queue length has been exceeded for incoming work. Once the work queue returns to levels below the set threshold, the cluster will dynamically release the added capacity and reform the cluster to its original static configuration.

Scalability

<table>
<thead>
<tr>
<th>“Classic” Microservices</th>
<th>Pneuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamically scale an individual service.</td>
<td>Pneuron supports sophisticated workload dispatching to deal with time-based variations in processing volumes.</td>
</tr>
<tr>
<td>Additional development or use of a third-party platform is required to monitor service execution, determine the need for scale-up, determine what resources are best utilized, accomplish the capacity addition and ultimately load balance requests across those multiple instances.</td>
<td>“Opt-in” specific virtualized “networks” and assign them to one or more specific Cortices (nodes), each with optional priority ranges assigned, to create the capability to segment incoming units of work and dynamically assign execution to a given “eligible / best” node.</td>
</tr>
</tbody>
</table>
service design and deployment capabilities with advanced execution functionality, Pneuron has eliminated the significant integration risk so common to the early adoption of new technology and paradigms.

Critically, Pneuron’s use of common standards within its own platform and in its communications with existing platforms, preserves and even enhances the value of those prior investments, and allows Pneuron to leverage existing best-of-breed functionality. Finally, **Pneuron’s non-invasive approach to integration minimizes disruption to ongoing operations and speeds solution deployment.**

**References:**

3 ) “Moving from the back office to the front lines”. IBM, 2014.
4 ) “Five Years From Now, CMOs Will Spend More on IT Than CIOs Do”. Forbes, February 8, 2012.

**About Pneuron**

Pneuron enables organizations to rapidly solve business problems through a groundbreaking, distributed approach that sidesteps the costly prerequisites of traditional approaches. Pneuron’s focus is on operationalizing the insights of “Big Data”, existing functionality of applications and the inherent potential of existing data through the deployment of intelligent workflows. These workflows are deployed as a non-invasive, distributed processing network that leverages a microservices-styled architecture. By visually configuring various native services (Pneurons) into a non-invasive “fabric”, designers combine data interactions, analytics, and process controls into intelligent workflows.