

# ESTUDO DE CASO – WINDOWS COMMUNICATION FOUNDATION DE ALTA PERFORMANCE NO GOVERNO DO ESTADO DE SÃO PAULO

## CASE STUDY - HIGH PERFORMANCE WINDOWS COMMUNICATION FOUNDATION IN THE GOVERNMENT OF THE STATE OF SÃO PAULO

Data de entrega dos originais à redação em: 21/12/2017  
e recebido para diagramação em: 06/04/2018

Eric Daniel David <sup>1</sup>  
Rogerio Homem da Costa <sup>2</sup>

O padrão de arquitetura orientado a serviços tem proporcionado aos serviços de fiscalização do governo grande poder de atuação e isso tem incrementado a preocupação com requisitos de segurança da informação. Neste cenário, como seria possível atender todas as diretrizes de segurança e ainda assim fornecer grandes massas de dados para execução das atividades de fiscalização externa? Este estudo de caso apresenta como este desafio foi vencido em um projeto deste tipo de órgão do governo do estado de São Paulo em que era necessário transmitir conjuntos de arquivos grandes.

Palavras-chave: SOA. WCF. Binding.

*The service-oriented architecture standarts has provided to government enforcement services with great power to act and this has increased concern about information security requirements. In this scenario, how would it be possible to meet all the security guidelines and still provide large masses of data for the execution of external oversight activities? This case study presents how this challenge was overcome in a project of this type on a department of the government of the São Paulo State where was necessary to transmit large sets of files.*

*Keywords: SOA. WCF. Binding.*

### 1 INTRODUÇÃO

As preocupações que órgãos públicos, em especial os serviços de fiscalização, têm com a segurança e o sigilo dos dados de seus contribuintes sempre fizeram parte dos pilares fundamentais do armazenamento de informações e tem importância destacada nestes órgãos. Por outro lado, com o passar do tempo tanto a evolução das ferramentas para a realização de cálculos e aferições destes dados, quanto a “expertise” dos utilizadores destas ferramentas, antes considerados “comuns”, ou ainda pouco especializados, avançou com a disseminação de novas tecnologias e a simplificação, principalmente, de aplicativos de análise de dados. Consequentemente a demanda de acesso a arquivos sigilosos por parte dos agentes de fiscalização aumentou e os riscos inerentes à segurança da informação se tornaram maiores.

O Decreto 3.724, de 10 de janeiro de 2001, dispõe sobre a requisição, acesso e uso das informações referentes a operações e serviços financeiros e, por consequência, dos dados dos contribuintes, no sentido de impor mecanismos de segurança, aplicados às informações destes contribuintes. Um órgão público do Estado de São Paulo aprimora a segurança das informações por meio da criação de camadas de serviços entre a aplicação e os dados armazenados. Com isso distingue-se o acesso e aprimora-se o controle de acesso à camada final de dados armazenados nos bancos de dados deste órgão, com vistas a mitigar acessos indevidos promovendo a proteção dos dados existentes em seus servidores e estimulando a integração de sistemas através de

serviços, pela combinação de vários serviços antes de tornar disponível a oferta final de informações a outros sistemas ou interfaces.

Duan (2005, p.1) diz que se comparada com as tradicionais arquiteturas distribuídas orientadas a objeto, a SOA (*Service Oriented Architecture*) mostra ser a mais adequada para integração sistêmica, quando envolver fatores como heterogeneidade e um ambiente de constantes mudanças. No passado, ao se falar em Sistemas Distribuídos, na prática, eram frequentemente distribuídos entre diversos nós de uma rede de computadores em uma única empresa ou em um grupo de parceiros de negócios próximos entre si. Hoje entende-se que vivemos um novo patamar desta distribuição, uma vez que várias empresas podem compartilhar *datacenters* e servidores usando computação em nuvem, criando softwares modulares distribuídos, que, apesar de serem soluções independentes, convergem para uma solução combinada e integrada (MERCURI, 2008, p.2).

A computação em nuvem é um ambiente onde são compartilhados recursos e serviços, onde as aplicações são disponibilizadas via internet através de hardware e software reconfiguráveis oferecidos como serviço por grandes *datacenters*. (LUCAS, 2017)

Este estudo de caso apresenta como a configuração e a implementação adequada de *bindings* do WCF (*Windows Communication Foundation*), parte vital desta tecnologia (PATEL, 2017), contribuiu para aprimorar o mecanismo de entrega de arquivos,

1 - Pós-Graduando em Gestão de Sistemas de Informação – IFSP Guarulhos. <ericdanieldavid@hotmail.com >.

2 - Professor da Pós-Graduação em Gestão de Sistemas de Informação – IFSP Guarulhos. <rogerio.costa@ifsp.edu.br >.

contendo dados sigilosos de contribuintes, para fiscalização externa em um órgão público do Estado de São Paulo.

## 2 DESENHO DA SOLUÇÃO

Para execução das atividades de fiscalização, os agentes de fiscalização dispõem de dois meios distintos para obter acesso às informações sigilosas dos contribuintes. O primeiro diz respeito ao acesso direto ao arquivo de dados (texto puro / bruto) e o segundo consiste em obter estes arquivos por meio de uma ferramenta computacional, desenvolvida por uma equipe de sistemas interna, na qual o arquivo é processado para um o agente de fiscalização que obtém as informações que deseja exibidas na forma de relatórios. Estes arquivos estão armazenados em tabelas de um banco de dados em formato *blob* (*Binary Large Object*).

A arquitetura de solução, como pode ser observada na Figura 1, seguiu os padrões sugeridos pela Microsoft, provedora das ferramentas de desenvolvimento, através de um guia focado no desenvolvimento de serviços para o governo do Estado de São Paulo.

Serviços *web* podem ser acessados basicamente por dois tipos de clientes: internos e externos. Os serviços internos, por estarem disponibilizados na intranet, não oferecem grandes desafios no que concerne à segurança pois ele está disponível do lado de dentro da empresa, têm acesso direto ao banco de dados e demais recursos indispensáveis a que ele necessite. (PAULA, 2013).

Importante ressaltar a restrição de comunicação com sistemas externos à rede deste órgão, que por sua vez apenas podem utilizar protocolo *http*, ou sua forma segura o *https*, para acesso a informações sistemicamente disponibilizadas.

Para definir os modelos de comunicação entre o serviço e seus consumidores o WCF utiliza os *bindings*. Cada *binding* determina qual canal será usado para comunicação e transmissão de uma mensagem entre os pontos de comunicação envolvidos.

Os canais são compostos por uma série de definições sobre a comunicação, a saber:

- Protocolo de comunicação utilizado:
  - HTTP;
  - Fila de Mensageria (MSMQ);
  - Pipes Nomeados;
  - TCP;
- Requerimentos de segurança;
- Sessões de confiança;
- Transações;
- Forma de codificação da mensagem:
  - XML;
  - Binário;
  - Message Transport Optimization Mechanism – MTOM; e
  - Outros.

As recomendações da Microsoft para utilização dentro do ambiente deste órgão público, levando em conta os servidores envolvidos e a estrutura de rede utilizada, se concentram nos seguintes *bindings*:

Tabela 1 – Diferentes bindings do WCF Fonte: Adaptado de PATEL, 2017

Binding	Descrição	Protocolo
BASICTHTPBINDING	COMPATÍVEL COM ASP.NET WEB SERVICE (*.ASMX)	HTTP/HTTPS
WSHTTPSBINDING	MENSAGENS CONFIÁVEIS, PROVENDO TRANSACIONALIDADE E SEGURANÇA UTILIZANDO OS MAIS NOVOS PADRÕES (WS*)	HTTP/HTTPS
WSDUALHTTPBINDING	CANAL DE COMUNICAÇÃO DUPLEX (DE DUPLO SENTIDO)	HTTP/HTTPS
WSFEDERATIONBINDING	PROVÊ SUPORTE PARA SEGURANÇA FEDERADA E PADROES DE SEGURANÇA WS*	HTTP/HTTPS
NETTCPBINDING	UMA CONEXÃO ORIENTADA AO USO DE TCP PURO	NET.TCP
NETNAMEDPIPEBINDING	UM CANAL DE COMUNICAÇÃO COM SUPORTE PARA PROCESSOS INTERNOS	NET.PIPE
NETMSMQBINDING	SUPORTE PARA COMUNICAÇÃO ATRAVÉS DO MICROSOFT MESSAGE QUEUE (MSMQ)	NET.MSMQ

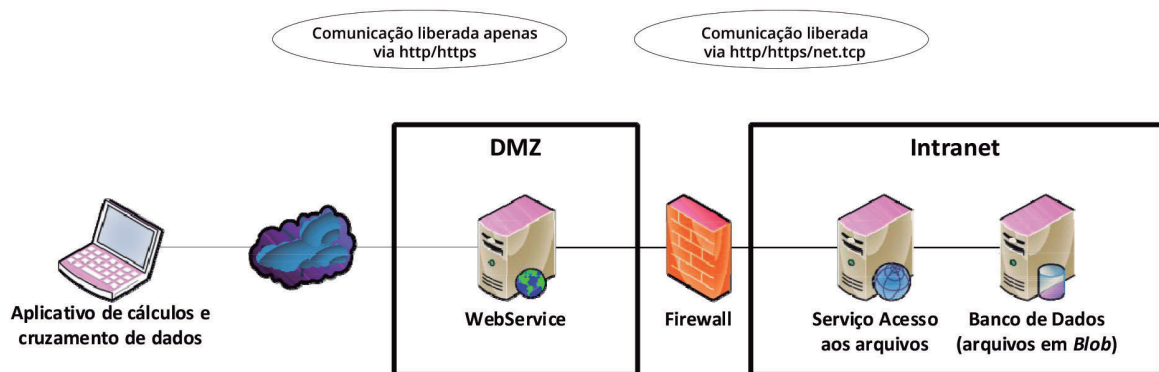


Figura 1 – Arquitetura da solução. Fonte: o autor

Tabela 2 – Diferentes tipos de Bindings disponibilizados pelo WCF. Fonte: Adaptado de PAULA, 2013

<b>Binding</b>	<b>Descrição</b>	<b>Desempenho aprimorado?</b>	<b>Interoperável?</b>
BASICTTPBINDING	EXTRANET	NÃO	SIM
WSHTTPBINDING WSDUALHTTPBINDING WSFEDERATIONHTTPBINDING	EXTRANET	NÃO	SIM (SOAP 1.1 E 1.2)
NETTCPBINDING NETNAMEDPIPEBINDING	INTRANET	SIM	NÃO
NETMSMQBINDING	PROCESSAMENTO BASEADO EM FILA, PROCESSAMENTO ASSÍNCRONO, CONFIABILIDADE, DISPONIBILIDADE	SIM	SIM (SOAP 1.1 E 1.2)

Dentre os requisitos não funcionais da solução podemos listar:

- Utilização por aproximadamente 1.000 agentes de fiscalização;
- Tamanho médio de arquivo 300 Mb;
- 30 arquivos solicitados por mês por agente fiscal.

Inicialmente a expectativa que a equipe de desenvolvimento tinha era que a carga neste mecanismo seria pequena, então optou pela utilização do *binding* *wsHttpBinding*, mantendo a interoperabilidade alta, possibilitando também a execução de teste diretamente pelo navegador das estações de trabalho.

Com isso em mente, e tendo sido todo o mecanismo implantado em ambiente de produção, agendou-se com a equipe de fiscalização externa o primeiro ciclo de avaliação, que consistia em fazer a obtenção de 4,65 Gb de arquivos através da solução completa, ou seja, do download dos arquivos através da porção cliente da ferramenta computacional, que estava na rede externa. Naquele primeiro ciclo constatou-se um tráfego total no adaptador de rede de 6,42 Gb no recebimento e 92,0 Mb no envio, aparentemente devido ao alto grau de informações de metadados do protocolo HTTP. Ou seja, foram 1,77 Gb (6.42 – 4.65) que consumiram a rede apenas para manter o controle do tráfego.

Este desperdício de banda de rede era grande. Após análise e seguindo a indicação da Microsoft, foi feita a alteração da configuração para utilização do *binding* no padrão *NetTcpBinding* na comunicação da DMZ com a *Intranet*, considerado, então, de melhor desempenho.

Cabe observar, que esta alteração foi possível apenas pelo fato de que todos os sistemas envolvidos haviam sido desenvolvidos utilizando-se tecnologia Microsoft .NET, o que permitiu a utilização de dados serializados usando codificadores binários, que, por sua vez, produziram dados serializados compactados (ASAD, 2017).

Essa alteração resultou em aumento da velocidade, uma vez que menos metadados eram transferidos. Entretanto a solução ainda não se mostrava ideal. Apesar da diminuição do tempo ter sido na ordem de

20% e da redução de desperdício de tráfego de 87% (apenas 237Mb de metadados foram transferidos, contra 1,77Gb já mencionado no teste com HTTP), o monitoramento dos servidores envolvidos na solução detectou que houve um aumento expressivo no uso de CPU.

Este aumento de uso de CPU justificou-se no fato do *binding* *NetTcpBinding* envolver o uso de algoritmos de criptografia da mensagem a ser transmitida, mecanismo obrigatório para realização de autenticação de serviços neste modelo. O algoritmo padrão usado nessa criptografia é o AES256, que trabalha com criptografia assimétrica, ou seja, utiliza um par de chaves, sendo uma chave pública e uma chave privada, ambas de 256 bits, para realizar a codificação da mensagem a ser transmitida.

A preocupação com o desempenho de *Web Services* seguros é legitimada pelo fato de que as especificações de segurança aumentam consideravelmente o tamanho da mensagem SOAP, principalmente o cabeçalho. Além disso, a adição dos elementos XML relacionados à segurança acarreta não apenas maior consumo de banda da rede para o transporte das mensagens SOAP, mas também consumo adicional de CPU para o processamento do documento XML e das operações necessárias à sua segurança. (RODRIGUES, 2011)

Apesar deste algoritmo de criptografia não ser o mais complexo e também de não usar chaves muito grandes, parte da equipe de desenvolvimento envolvida cogitou a utilização de chaves de 128 bits para tentar reduzir o uso de CPU já que a segurança estaria balanceada suficientemente (SINGLETON, 2016).

Avaliou-se, porém, que ainda não era uma solução ideal para o mecanismo uma vez que, nesta bateria de testes apenas 1 cliente com 3 threads estava sendo executado, não sendo esta uma solução escalável passível de ser usada para atender os 1.000 agentes de fiscalização previstos. Um resumo das mensurações coletadas nesta primeira bateria de testes pode ser observado na tabela 3 abaixo:

Tabela 3 – Primeira avaliação. Fonte: o autor

<b>Binding Externo</b>	<b>Binding Interno</b>	<b>Consumo de rede</b>	<b>Perda</b>	<b>CPU média</b>
WSHTTPBINDING	WSHTTPBINDING	6,42 GB	1,77 GB	15%
WSHTTPBINDING	NETTCPBINDING	4,89 GB	0,23 GB	60%

## 2.1 Otimização através da reconfiguração

Após alguns estudos a equipe de desenvolvimento notou que haviam alguns tipos de *binding* do WCF que não foram considerados pela Microsoft em suas recomendações. Um dos pontos que mais chamou atenção foi que alguns destes *bindings* foram disponibilizadas no Microsoft .NET Framework em versões posteriores (4.5) à versão utilizada do guia de recomendação (4.0).

Um destes novos *bindings*, em especial, atraiu atenção após feita leitura de sua documentação, pois prometia o melhor de dois mundos: comunicação por protocolo HTTP, mas com dados serializados de forma binária, uma de suas restrições é que os clientes deveriam usar Windows 8 ou posterior (NIEMI, 2013).

Iniciou-se assim a implementação deste novo *binding*, o *NetHttpsBinding*, entre todas as camadas de serviço, ou seja, não apenas na comunicação da DMZ com a *Intranet*, mas também da DMZ com o cliente externo.

Utilizando-se dos mesmos mecanismos de segurança, ou seja, criptografia na mensagem através do algoritmo AES256, a avaliação do mecanismo em ambiente de desenvolvimento foi extremamente surpreendente, sugerindo que o desempenho em produção agregaria aumento da *performance* de forma muito vantajosa.

## 3 RESULTADOS

Para a bateria de testes finais foram executadas as extrações usando 10 threads, visando ampliar o uso do CPU simulando o uso mais realista da aplicação porém sem comprometer as demais aplicações que utilizam o mesmo servidor. Os testes foram executados em horário em que as demais aplicações não teriam carga de trabalho, portanto a possibilidade de discrepância por interferência foi mínima.

A Figura 3 mostra o comportamento do CPU, deixando claro que a adoção deste *binding* do *Windows Communication Foundation* proporcionou o desempenho aderente aos requisitos não funcionais do projeto.

A utilização, portanto, deste *binding* foi oficializada nos demais nós do *cluster* e efetivadas nos sistemas

disponibilizados ao usuário final. Desde então, não houve incidentes abertos por usuários relacionados ao desempenho do sistema.

O monitoramento do ambiente, feito pela equipe do Centro de Operações de Infraestrutura deste órgão público do Estado de São Paulo, não tem apresentado grandes oscilações na demanda de CPU dos servidores envolvidos. De toda a infraestrutura envolvida atualmente não foram identificadas novas oportunidades de melhoria (otimização) na camada de serviços, apenas no banco de dados, tendo em vista que a rede do servidor de banco de dados está limitada a 100 mbps, bem como pelo fato de que as tabelas que contém os *blobs* de arquivos não estarem particionadas.

Uma ideia para testes futuros consiste na utilização do ASP.NET Web API, por ser mais simples do que o WCF. Um dos principais argumentos para esta alteração é a complexidade das configurações do WCF, o que torna sua manutenção “não atraente” aos desenvolvedores, tendo em vista que o formato JSON (JavaScript Object Notation), tendência mais atual em relação do transporte de dados em relação ao formato XML, tem atraído muita atenção e dedicação de estudo por parte dos desenvolvedores (SINGLETON, 2016). Alguns estudos também apontam que serviços em ASP.NET Web API tem melhor desempenho já que não transportam todo cabeçalho SOAP (SHRINGARE, 2017) sendo este seu possível grande ganho.

## REFERÊNCIAS

- BRASIL. DECRETO Nº 3.724, DE 10 DE JANEIRO DE 2001.
- DUAN, Z; BOSE, S; SHONIREGUN, C. SOA Without Web Services: a Pragmatic Implementation of SOA for Financial Transactions Systems. **IEEE International Conference on Services Computing**, 2005.
- MERCURI, M. Considerations for Designing Distributed Systems. **The Architecture Journal**, Seattle, USA, n.17, p.2-5, oct. 2008.
- PAULA, H. SP **Guia de Desenvolvimento para Serviços Web**. Microsoft, 2013

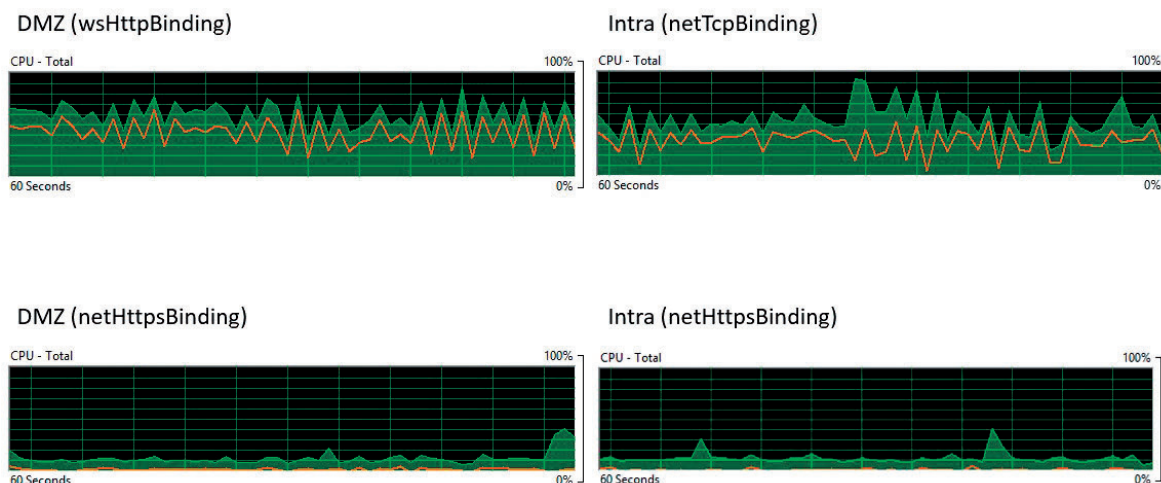


Figura 3 – Comparativo de utilização de CPU (NetTcpBinding VS NetHttpsBinding). Fonte: o autor

RODRIGUES, D; ESTRELLA, J. C.; BRANCO, K. RLJC. **Avaliação de Desempenho de Algoritmos Criptográficos em Web Services Utilizando WS-Security**. 2011.

PATEL, Chirag (Ed.). **Developing Service-Oriented Applications Using the Windows Communication Foundation (WCF) Framework**. IGI Global, 2017.

ASAD, Ali; ALI, Hamza. Serialization and Deserialization. In: **The C# Programmer's Study Guide (MCSD)**. Apress, 2017. p. 305-318.

SINGLETON, James. **ASP.NET Core 1.0 High Performance**. Packt Publishing Ltd, 2016.

SHRINGARE, Komal. SOA Approaches Analysis and Integration with WCF Services. **AADYA-National Journal of Management and Technology (NJMT)**, v. 7, p. 207-218, 2017.

NIEMI, Jarno et al. **WCF-palvelut & Windows Azure Cloud Service**. 2013.

LUCAS, Thiago Jose et al. COMPUTAÇÃO EM NUVEM-DISPONIBILIDADE: PESQUISA APLICADA NA FACULDADE DE TECNOLOGIA DE OURINHOS. **RETEC-Revista de Tecnologias**, v. 9, n. 2, 2017.