1710 - Defining software technology			
BENCHMARKS	**ACTIVITIES BY YEAR**		
	2017		
	'1-1	'2-1	'3-1
(none)	Big data examples	Technology vs. features	Ineligible activities
OBJECTIVES	**RESULTS**		
Reliability: 99 %	88		
Scalability: 8500 %	4400		
Availability: 99.9 %	99.3		
UNCERTAINTIES & KEY VARIABLES	**CONCLUSIONS**		
1 - O/S design issues			
kernel - process, memory & I/O mgmt	Y		
process mgmt - scheduler lists & queues	Y		
2 - Micro processors			
compilers		Y	
Interface		Y	
linkers		Y	
programming languages		Y	
3 - Other areas			
big data methods (store, process, manage)			
improved infrastructure options			
scaling, reliability & availability options			
technology stack or tool designs			
vision design options			
	METHODS		
Analysis	250	147	
Trials	17	13	
Prototypes		7	
Lines of code		14800	
	COSTS		
Hours	100	250	
Materials $			
Subcontractor $			

SREDStakeholder.CA

Project Name:	Defining software technology	**Start Date:**	2017-01-02
Project Number:	1710	**Completion Date:**	2018-03-21

Project Details:

Scientific or Technological Objectives:

Measurement	Current Performance	Objective	Has results?
Reliability (%)	65	99	Yes
Scalability (%)	100	8500	Yes
Availability (%)	96	99.9	Yes

Canada Revenue Agency (CRA) Eligibility of work in SR&ED projects containing software development - Transcript Mach 22, 2017

Slide 10:

As mentioned in an earlier slide, Experimental Development is undertaken for the purpose of achieving technological advancement.

So, what is technology?

Technology is not a physical entity.

Technology is the practical application of scientific knowledge and principles.

It is the knowledge of how scientifically determined facts and principles are embodied in the material, device, product, or process.

As technology evolves, the practical applications of scientific knowledge and principles as well as the technological knowledge associated with the products, processes, or devices become the scientifically determined facts and principles for further development.

Let’s look at some examples to illustrate this.

Field of Science/Technology:

Software engineering and technology (2.02.09)

Project Details:

Intended Results:	Develop new processes, Improve existing processes
Work locations:	Lab
Key Employees:	
Evidence types:	Progress reports, minutes of project meetings; Test protocols, test data, analysis of test results, conclusions; Records of resources allocated to the project, time sheets; Samples, prototypes, scrap or other artefacts; Design, system architecture and source code; Project records, laboratory notebooks; Project planning documents; Photographs and videos; Design of experiments; Records of trial runs; Contracts

Scientific or Technological Advancement:

Uncertainty #1: O/S design issues

Slide 11: Example 1:

A microcontroller or microprocessor can run machine code without an operating system and any user developed software can run directly on it.

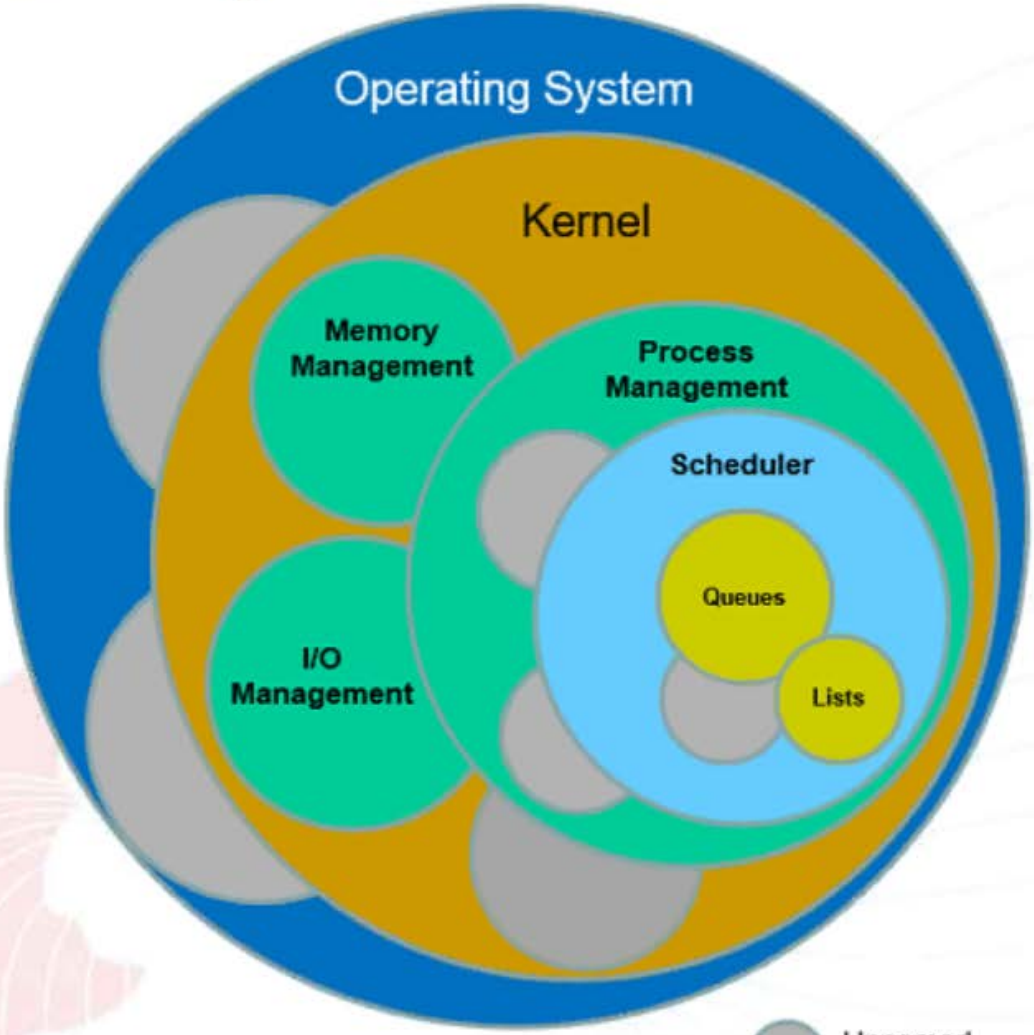
SREDStakeholder.CA

In contrast, in an Operating System, user developed software runs within a managed environment.

The kernel has process management, memory management, and input/output (I/O) management to govern the environment.

Technology – Example 1





Within process management, a scheduler manages tasks and processes using complex data structures working in conjunction with other embodiments. As we can see, within a modern Operating System, kernel and other embodiments not mentioned in the slide make up the scientifically determined facts and principles.

Similarly for the kernel, it is the process management, memory management, I/O management and others. As we go down further, the scheduler has Lists, Queues, etc.

The most significant underlying key variables are:

kernel - process, memory & I/O mgmt, process mgmt - scheduler lists & queues

Technology or Knowledge Base Level:

Activity #1-1: Big data examples (Fiscal Year 2017)

Methods of experimentation:

Method	Experimentation Performed
Analysis / simulation:	250 alternatives
Trials:	17 runs / samples

Slide 13:

We looked at the definition of technology through software examples.

New products/services are being built using or combining various technologies for creating complex applications. This is driving relentless progress.

The trend in software development has shifted in the direction of shared services such as Software as a Service, Platform as a Service and Infrastructure as a Service.

We see this manifest as Cloud computing which is made possible by tremendous advancements in hardware and software technologies.

From software perspective, Cloud services became possible through advancement in networking and distributed infrastructure and computing.

All these developments make use of various advancements in software technology stack.

This is an illustration of how various advancements provide a platform for modern big data applications.

Results:

- Reliability: 88 % (67% of goal)
- Scalability: 4400 % (51% of goal)
- Availability: 99.3 % (84% of goal)

Conclusion:

As you all know, different companies compete in different areas of technology. For example, a company working on scalable distributed computing and processing deals with different technologies than a company working on databases or content delivery network. These companies have the potential of advancing technology in their respective areas.

Significant variables addressed: kernel - process, memory & I/O mgmt, process mgmt - scheduler lists & queues

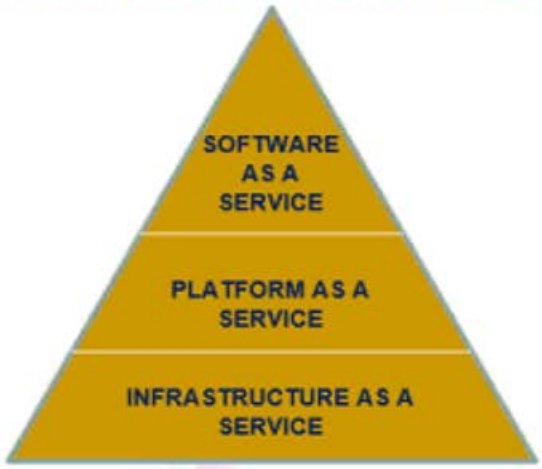
Uncertainty #2: Micro processors

SREDStakeholder.CA

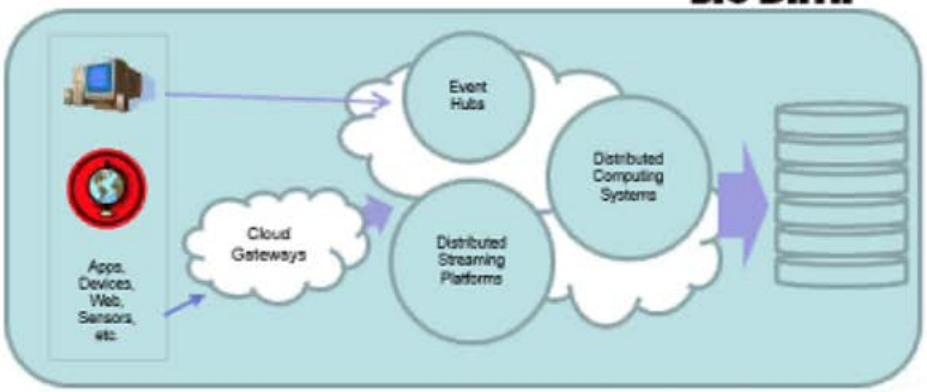
Slide 12:

Even when we use a microcontroller without an Operating System, the tools used to generate the machine code can come from a tool-set.

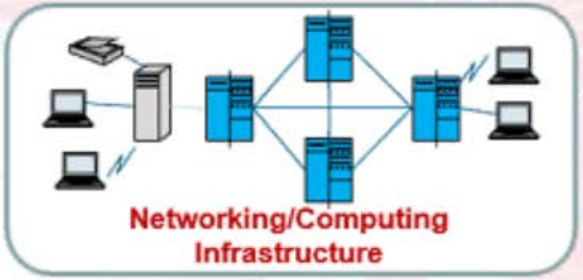
Various advancement areas



BIG DATA







Technology Stack

- Languages
- Content Management Services
- Content Delivery Network
- Caching
- Applications
- Databases
- Operating Systems

For example, a well known open source electronic platform provides a tool which uses the principles of programming language to enable writing in a higher level language, uses the principles of compiler that turns the code into object files (machine readable instructions), and a linker to combine the object files with libraries.

Principles of programming languages, compiler, linker, and Graphical User Interface are some of the scientifically determined facts and principles within the tool-set.

Similar evolution can be seen in technologies associated with Web systems, Information systems, micro-controller based systems, relational databases, No-Structured Query Language (SQL) databases, distributed storage and retrieval etc.

The most significant underlying key variables are:

programming languages, compilers, linkers, Interface

Technology or Knowledge Base Level:

Activity #2-1: Technology vs. features (Fiscal Year 2017)

Methods of experimentation:

Method	Experimentation Performed
Analysis / simulation:	147 alternatives
Trials:	13 runs / samples
Physical prototypes:	7 samples
Lines of code:	14800 Lines of prototype code

Slide 14:

Companies produce various products and services using technology. SR&ED looks at the advancement in technology necessary to deliver the new or improved products and services. Therefore it is important to distinguish between technology and product.

Often the name of the product, process, or device is also used to identify the technology.

The distinction is:

Product, process, or device conveys feature, functionality, capability, etc.

Technology is about the knowledge of how the constituents embodied within the product, process or device, function to deliver the feature, functionality, capability, etc.

For example: Web information system publishing tools that also serve as document management and storage systems come with features, interfaces, application program interfaces (APIs) and tool-sets to build information systems.

Results:

Conclusion:

Slide 15:

Understanding the capabilities and limitations of such systems for building an information system, use of their API/interfaces to develop an information system, or information about their features and functionality are about a product.

On the other hand, knowledge of the internal workings of such systems and how inter-relationships produce, influence, and impact the features, interfaces, APIs, tool-sets etc., are related to the technology.

Significant variables addressed: compilers, Interface, linkers, programming languages

Documentation:

- Offline Documents: Supporting documentation

SREDStakeholder.CA

Uncertainty #3: Other areas

Slide 48:

Technological advancements in software development projects can involve:

New or improved techniques and methods in computing to store, search, process, and manage vast collection of data (big data). Typically undertaken in Universities and large research labs.

Advancements necessary in support of software technology stack or tools.

Advancements necessary for new or improved infrastructures like internet driven Cloud and distributed computing.

Advancements necessary to support scaling, reliability and availability of software based systems.

Advancements in other technology areas like vision and medical imaging, video transmission, telecommunication (voice), automation, controls, etc.

The most significant underlying key variables are:

improved infrastructure options (unresolved), scaling, reliability & availability options (unresolved), vision design options (unresolved), technology stack or tool designs (unresolved), big data methods (store, process, manage) (unresolved)

Technology or Knowledge Base Level:

Activity #3-1: Ineligible activities (Fiscal Year 2017)

Methods of experimentation:

Slide 27:

We will now look at a few examples to illustrate common misconceptions about technological uncertainty. The first example is to emphasize that absence of a capability does not equate to a technological uncertainty.

A company wants to build a new version of their client server mobile application. The older version of the application worked in conjunction with a well known Enterprise Server that offered server side Push. The new version of the application was required to work in setups where the Enterprise server could not be used.

In client server applications, the server responds to client requests. With a server side Push service, client applications can receive information from their servers even if they did not initiate a request.

Results:

Conclusion:

The company cannot claim the lack of availability of the Push capability from the Enterprise server as a technological uncertainty for its work to build some form of a Push service.

The company has to identify a limitation of technology that is preventing the development of its own Push service.

SREDStakeholder.CA

1711 - VOIP - improve SIP interaction	
BENCHMARKS	**ACTIVITIES BY YEAR**
Internet searches: 32 Articles Patent searches: 5 patents Competitive products or processes: 17	2017
Potential components: 8 products	'1-1
Queries to experts: 5 responses	Activity 1
OBJECTIVES	**RESULTS**
Call continuity: 99.9 % uptime	94
UNCERTAINTIES & KEY VARIABLES	**CONCLUSIONS**
1 - TU beyond SP	
Information theory	Y
Network protocols-TCP, SIP, RTP	Y
VOIP protocols - setup, progress, teardown	Y
	METHODS
Analysis Trials Prototypes Lines of code	
	COSTS
Hours	170
Materials $	
Subcontractor $	

SREDStakeholder.CA

Project Name:	VOIP - improve SIP interaction	**Start Date:**	2017-02-01
Project Number:	1711	**Completion Date:**	2017-12-31

Project Details:

Scientific or Technological Objectives:

Measurement	Current Performance	Objective	Has results?
Call continuity (% uptime)	85	99.9	Yes

Slide 17:

It is important to establish the technological knowledge base to identify a technological uncertainty.

Let us look at an example to illustrate the kind of information useful in establishing the technological knowledge base.

As an example, we can think of a telecom software company developing Automatic Call Distributors (ACD) for Voice over Internet Protocol (VoIP) based voice telephony, wants to prevent failures in Session Initiation Protocol (or SIP) calls due to hardware, software, system, or network failures.

Peer to peer connections through data network are susceptible to such failures resulting in session disruptions.

For voice sessions over data, this has the undesirable effect of loss of on-going phone calls and call setup in progress.

Field of Science/Technology:

Telecommunications (2.02.07)

Project Details:

Intended Results:	Improve existing processes
Work locations:	Lab
Key Employees:	
Evidence types:	Project planning documents; Design of experiments; Records of trial runs

Scientific or Technological Advancement:

Uncertainty #1: TU beyond SP

Slide 18:

This company asserts that the objective of call continuity (continuing phone calls without re-initiating them), cannot be achieved based on the technological knowledge base.

The information used by the company in coming to such a conclusion can help in establishing the technological knowledge base:

Starting with identifying the technology or technologies involved (such as session initiation protocol, session description protocol (SDP), real-time transport protocol (RTP) and their interactions);

identifying what is available in the technological knowledge base with regard to peer to peer calling, failure modes, call set-up, call progress, and call tear-down;

describing any proprietary technology involved;

Slide 19:

identifying any patents or technical information associated with how call failures are handled using the current state of technology;

providing the technological experience of the company in the technology areas such as VoIP, Information theory, Voice telephony, Network protocol knowledge of Transmission Control Protocol (TCP)/User-Datagram Protocol (UDP)/SIP/SDP/RTP, etc.;

SREDStakeholder.CA

identifying where the current state of technology is insufficient; and

having this research documented when it was done in order to establish the technological knowledge base at that specific point in time.

Establishing the technological knowledge base properly before the assertion of a technological uncertainty will avoid re-inventing the wheel.

The most significant underlying key variables are:

Network protocols-TCP, SIP, RTP, Information theory, VOIP protocols - setup, progress, teardown

Technology or Knowledge Base Level:

Benchmarking methods & sources for citings:

Benchmark Method/Source	Measurement	Explanatory notes
Internet searches	32 Articles	identifying what is available in the technological knowledge base with regard to peer to peer calling, failure modes, call set-up, call progress, and call tear-down;
Patent searches	5 patents	identifying any patents or technical information associated with how call failures are handled using the current state of technology;
Competitive products or processes	17 products	Starting with identifying the technology or technologies involved (such as session initiation protocol, session description protocol (SDP), real-time transport protocol (RTP) and their interactions);
Similar prior in-house technologies	3 products / processes	describing any proprietary technology involved;
Potential components	8 products	discuss limits of available technologies with vendors
Queries to experts	5 responses	identifying where the current state of technology is insufficient

Activity #1-1: Activity 1 (Fiscal Year 2017)

Methods of experimentation:

Slide 33:

Once a technological uncertainty is identified, the techniques used for systematic investigation or search can be techniques that are within the technological knowledge base.

Let us illustrate the last point with the example of experimentations that require the monitoring of data traffic in networks through packet capture using packet sniffing and analysis tools.

Even though the technique of capturing and monitoring data traffic through packet sniffing and packet analysis may be well known and are considered to be in the technological knowledge base, they are essential for experimentations where the results of the experiments are derived from the information in the packets.

Results:

- Call continuity: 94 % uptime (60% of goal)

Conclusion:

Significant variables addressed: Information theory, Network protocols-TCP, SIP, RTP, VOIP protocols - setup, progress, teardown

Documentation:

- Offline Documents: Documentation

SREDStakeholder.CA

1712 - Software security example	
BENCHMARKS	**ACTIVITIES BY YEAR**
	2017
	'1-2
(none)	Eligible activities
OBJECTIVES	**RESULTS**
Single sign on: 1 1 Yes / No 0	1
Use of external service: 0 1 Yes / 0 No	0
Reliability: 99.9 %	99.9
Monthly cost: 500 $	740
Log on time: 1.5 seconds	0.6
UNCERTAINTIES & KEY VARIABLES	**CONCLUSIONS**
1 - Technological uncertainty	
DS user object attributes (where & how)	
how to map security tokens	
relation to cloud objects	
	METHODS
Analysis	475
Trials	23
Prototypes	
Lines of code	
	COSTS
Hours	450
Materials $	
Subcontractor $	

SREDStakeholder.CA

Project Name:	Software security example	**Start Date:**	2017-03-01
Project Number:	1712	**Completion Date:**	2018-06-20

Project Details:

Scientific or Technological Objectives:

Measurement	Current Performance	Objective	Has results?
Single sign on (1 Yes / No 0)	1	1	Yes
Use of external service (1 Yes / 0 No)	1	0	Yes
Reliability (%)	99.9	99.9	Yes
Monthly cost ($)	5000	500	Yes
Log on time (seconds)	4	1.5	Yes

Slide 21:

As you remember from slide 16, we discussed technological uncertainty and the importance of identifying the shortcomings or limitations of technology. Next, we will be looking at limitation of technology through an example.

Here are some background information in preparation for the example:

As we are aware, access to software applications is based on confirming the identity of the user. Software applications can rely on other identity providers and trust the ID given by these providers. This is very similar to how organizations like banks rely on government issued drivers license for identification.

When there are many applications, it is convenient to be able to use the same ID for accessing them. It is even better if the process of verification for access control is performed in the background through pre-defined security policies. This type of access to software applications is referred to as single sign-on. The security of communications to request, generate, and provide ID are critical to the integrity of single sign-on.

The illustration shows one such security trust service model defined by a consortium of companies. A user requiring access to an application obtains security token from an ID provider that is trusted by the relying applications. These security tokens contain codified claims to protect and authorize requests. The claims are assertions about the user.

Web Services-Federation referred to as WS-Federation is an evolution of this model. It addresses the needs of web applications with richer trust relationships and advanced federation of services.

Slide 22:

Now, let us look at limitation of technology through an example related to single sign-on

In 2013, a leading software company XYZ offered Cloud based access to its productivity software suite and related services through single email login authentication.

Its single sign-on (SSO) login capability was constructed on claims based user identity using its Cloud based directory and identity management service.

Field of Science/Technology:

Project Details:

Intended Results:

Work locations:

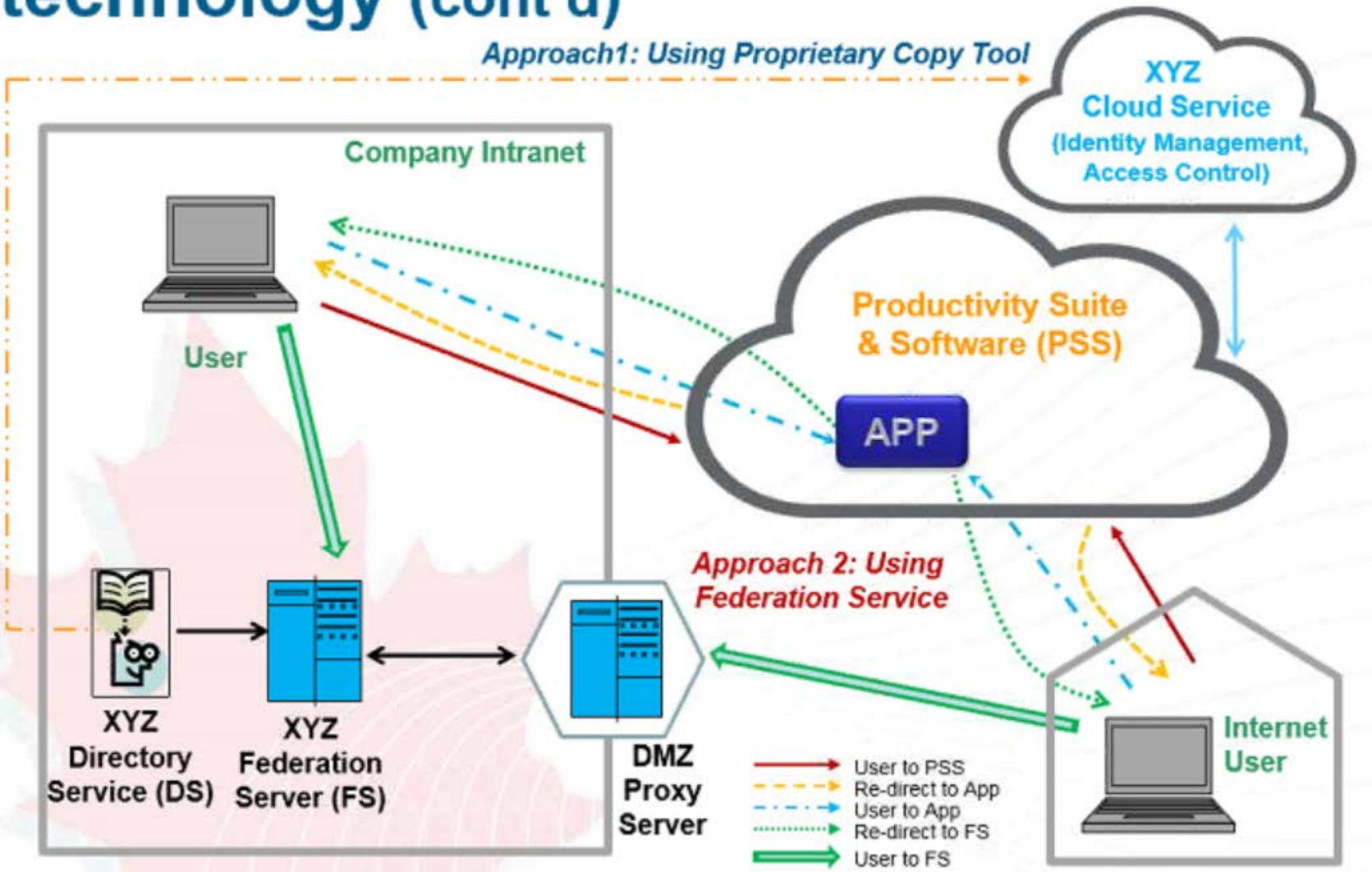
Key Employees:

Evidence types: Test protocols, test data, analysis of test results, conclusions; Records of resources allocated to the project, time sheets; Design, system architecture and source code; Project planning documents; Records of trial runs

SREDStakeholder.CA

COMMERCIAL CO

Example: Identifying a limitation of technology (cont'd)



Scientific or Technological Advancement:

Uncertainty #1: Technological uncertainty

Slide 23:

At that time, there were two approaches generally available in the technological knowledge base to implement single login authentication for applications offered through the Cloud services of the XYZ software company. They were as follows:

The company offered a proprietary tool to synchronize the essential information from an organization's internal directory service to its Cloud based directory and identity management software. This will provide local users of the organization, access to the applications hosted in the Cloud from both inside and outside the office.

Setting up a Federation Service: The company also offered a Federation Service software for claims based access control authorization which allowed single sign-on for multiple security realms (domains). It functioned as a Security Token Service relying on individual domain controllers and performed authentication and provided security tokens for WS-Federation. This Federation Software could be used with its Cloud-based directory service for single sign-on through trust relationship.

Slide 25:

Both these approaches discussed in the previous slide, impose restriction on usernames through namespace hierarchies based on domain segregation. Additionally,

approach 1 is not a true single sign-on, rather a way to allow internal users to access from outside, while
approach 2 is a single sign-on but requires a federation server and proxy server for every domain controller. It also uses domain user accounts for login

As an example, a company project objective was to develop a single sign-on solution using the XYZ Cloud's identity and access control without having username restrictions or requiring the Federation Service for identity management.

Slide 26:

Now let us look at what is the limitation of technology in this example.

The federated single sign-on option available in the XYZ Cloud was using WS-Federation protocol.

The Cloud's identity management trusted the Federation Service to provide the security tokens.

The technological knowledge base, including understanding of how the XYZ Cloud's identity management worked, was insufficient for the company to develop a solution to overcome the restrictions imposed by approach 2.

At this stage, this is a limitation of the current state of technology.

Work can be undertaken to advance the knowledge towards the development of this solution that will enable single sign-on.

The most significant underlying key variables are:

DS user object attributes (where & how) (unresolved), how to map security tokens (unresolved), relation to cloud objects (unresolved)

Technology or Knowledge Base Level:

Activity #1-2: Eligible activities (Fiscal Year 2017)

Methods of experimentation:

Method	**Experimentation Performed**
Analysis / simulation:	475 alternatives
Trials:	23 runs / samples

SREDStakeholder.CA

Slide 43:

This example for scenario 3 is a continuation of the previous example under “Limitation of technology” and describes the work that was undertaken towards advancing the understanding of how XYZ Cloud uses information in the Directory Service object for single sign-on identify management.

Let us look at the work undertaken.

A setup consisting of XYZ Cloud’s Access Control Service using the Cloud’s Directory Service instance as the ID provider for a relying application was created. A packet sniffing tool was setup to capture and analyze the user browser network traffic and log during login attempts. The Cloud’s Directory Service was populated with user objects from a local Directory Service using the proprietary synchronization tool as in approach.

The visibility of claims within the WS-Federation was confirmed during login attempts. The next effort was to understand the relationships between claim attributes and Directory Service user objects. The observations and analysis of the claim attributes of the captured information within security tokens, allowed the mapping and expression of the correlation between the Directory Service user object attributes and the object attributes used within the Cloud’s identity management entities.

Slide 44:

The next activities were towards validating the relationships, determining whether Security Token Service provided by the Federation Service can be substituted, and finding out what may be necessary to mimic any Federation Service specific behavior.

To accomplish this, an open source security token server was setup. A trust association between the server and XYZ Cloud identity management was established. The signing algorithm and signing certificates were aligned between the two. These steps were necessary to configure the use of the security token server as the identity provider for the Cloud Service. Based on the developed relationships, the server source code was modified to create new claim types structured using Directory Service user object attributes.

By undertaking several tests, it was possible to understand where and how the Directory Service object attributes were used in the claims for the Cloud Service. The results enabled the construction of a single sign-on for XYZ Cloud Service without Federation Service where it was possible to use attributes and unique-ids that did not belong to the Directory Service. This also led to the development of user stores without the Directory Service.

Again, what conclusions can we draw from the work undertaken in this example?

Summary of documentation process 53-55:

Slide 53:

Confirmation of the visibility of Directory Service user attributes inside the claims within WS-Federation credentials was done using a packet sniffing and analysis tool and therefore the following results would have been naturally or easily produced:

Screen-capture or log of the responses identifying the tokens with claims, and the recognizable attributes of the Directory Service user objects with or without encoding.

Slide 54:

The next step to map the attributes to the claim identifiers should produce:

Names and format of expression of all the claim identifiers and their associated user objects in the claims
The trust association between the generic Security Token Server and XYZ Cloud was set-up and made functional. A screen capture of the response of the appropriate commands used to establish this association will provide the details of what was accomplished.

Slide 55:

Finally, code changes were introduced and modified in the generic Security Token Server to enable claim construction. Several tests were conducted and adjustments were made based on the results of the tests to enable successful claims and security tokens. This work should have:

initial modification to the source code to create claims and the results of using it, changes made based on the results of previous tests and their outcome, and the final code.

SREDStakeholder.CA

Results:

- Single sign on: 1 1 Yes / No 0 (100% of goal)
- Use of external service: 0 1 Yes / 0 No (100% of goal)
- Reliability: 99.9 % (100% of goal)
- Monthly cost: 740 $ (94% of goal)
- Log on time: 0.6 seconds (136% of goal)

Conclusion:

Slide 45:

From the work described, we can conclude that the Information on the work undertaken is at sufficient level of detail that both a claimant and a reviewer from the Canada Revenue Agency (CRA) can agree on the statements of facts, see the characteristics of the work undertaken, and identify new knowledge. With reasonable effort, the reviewer (and the claimant) can determine and agree if this is over and above the technological knowledge base.

Slide 46:

Here are some additional points on systematic investigation or search.

If the work generates or confirms what is within the technological knowledge base, even though the work was undertaken systematically, it will not be considered as SIS for reducing or eliminating technological uncertainty.

Work resulting in generation of new knowledge that explains the reasons of failure of possible solutions to achieve a technological objective is considered a technological advancement provided it is not available in the existing technological knowledge base.

Slide 51:

In this project, understanding
- where and how the Directory Service user object attributes are used in the claims,
- how they are mapped to the security tokens,
- their relationships to the Cloud identity management user object attributes and
- ruling out the need to mimic any Federation Service specific behavior

was the technological advancement.

Because this knowledge allowed the building of ‘single sign-on capability without ‘the need to use the XYZ software company’s Federation Service’ in the XYZ Cloud and the creation of user store not based on XYZ Directory Service for single sign-on.

It is important to note that achievement of single sign-on capability on its own is not the technological advancement.

Documentation:

- Offline Documents: Documentation

Uncertainty #2: Hypothesis (for SR&ED)

Slide 34:

After having discussed technological uncertainty, we will illustrate the concept of hypothesis which is directly relevant to answering Question 2.

Appropriate hypothesis can be formulated for further investigation after identifying what is lacking in the scientific or technological knowledge base.

So, what is hypothesis?

Hypothesis is an idea consistent with known facts.

For example, internet data transmission speed can exceed the speed of light is not consistent with known facts.

Hypothesis is a supposition or explanation on the basis of limited evidence which serves as the starting point for further investigation.

SREDStakeholder.CA

Example 1: While going through technology stack selection to decide between two real-time Operating System options, a supposition that real-time Operating System ‘A’ will offer better interrupt management than real-time Operating System ‘B’, has no linkage to anything lacking in the scientific or technological knowledge base. Therefore it is not a hypothesis for

SR&ED.

Example 2: Continuing on the previous example on limitation of technology,

a hypothesis can be that

- specific information from the Directory Service objects are
- used in the composition of claims for the Cloud service security tokens within WS-Federation.

This on the other hand, is a hypothesis for SR&ED.

Formulation of proper hypotheses is essential for planned approach to SR&ED.

The most significant underlying key variables are:

Directory service objects (unresolved), Cloud security tokens (unresolved)

Technology or Knowledge Base Level:

There are no Activities associated with this uncertainty.

SREDStakeholder.CA