

ca world®'11

IT at the speed of business

using CA IDMS™ in the cloud

agility
made possible™

ca
technologies

Next-Generation Mainframe Management

MI460SN

using CA IDMS™ in the cloud

Tom Hebert

ObjEx, Inc.

ca world®'11



*Experts in IDMS
Integration*

- Cloud Computing may be the best thing to happen to CA IDMS in many years. Your organization may be looking at Cloud options, but not fully understand where CA IDMS fits. What if you use software products that embrace SOA (Service Oriented Architecture)? What if you want to create new applications that use Cloud techniques and technology? How will you get there? With his expertise in SOA design and development and deep CA IDMS experience, Tom Hebert answers these questions and describes how you can wrap existing CA IDMS applications with an architecture that can take advantage of Cloud and other new technologies.

scalable cloud computing with CA IDMS?

introduction

- Cloud computing may be the best thing to happen to CA IDMS in many years
- How do you use your CA IDMS data to meet new integration demands?
 - Salesforce.com
 - Industry collaboration sites
 - Social networking sites
 - Marketplaces like Amazon.com and others
- How do you implement, securely, with performance - without having to rewrite your applications?

does scalable cloud computing with CA IDMS require a port? don't focus on your navel

- Relational is losing its monopoly on data
 - Consumers of cloud services do not care which database is being used
 - Witness the NoSQL movement
 - There is no advantage to using a relational database on the cloud
 - It is counterproductive to port everything before you can start work on things that are critical to your organization's success
- The way to move forward is to:
 - Keep CA IDMS
 - Allow current applications to live out their useful lives
 - Wrap CA IDMS with an architecture that allows you to use your CA IDMS data in your cloud

scalable cloud computing with CA IDMS?

introduction

- You aren't going to do it with screen scrapers
 - Imagine what it would be like building a real-time bridge with an outside sales provider, limiting yourself to your existing dialogs
- You aren't going to do it with your data warehouse
- It takes forethought, but it can be done
 - Software architecture
 - Organize for unknown requirements
 - Deployment architecture and plan
 - Scalable, reliable, and low-cost

cloud computing vs. “the cloud”

- “The cloud” is the collection of cloud computing services available on the internet
- Cloud computing is a design and implementation architecture
 - Allows us to safely invest in our applications
 - Typically built using a Service Oriented Architecture
 - Currently SOAP and TCP/IP prevails
- Your private cloud is the collection of services provided and maintained by your organization
 - Some are private
 - Some are exposed outside the organization

cloud computing defined

- You may not know it but you have been using the cloud
 - Bing, Google Maps, Salesforce.com, and many others
- Not just web sites
 - e.g., when Hertz or Marriott want to show you a map they don't send you to Google
 - True for airlines, travel sites, industry sites, and many others
- EDI will eventually go the way of the dodo bird

cloud computing defined

- Much more than merely putting up a web site
- Driven by major internet players
 - A fabric of interconnected callable services
 - Driven by users and consumers, not by the accounting department
- Interoperable and heterogeneous
 - Any platform, operating system, or database can play as long as they follow the rules
- Documents rule (levels the playing field)
 - Relational client/server model not effective
 - XML allows complex requests and responses

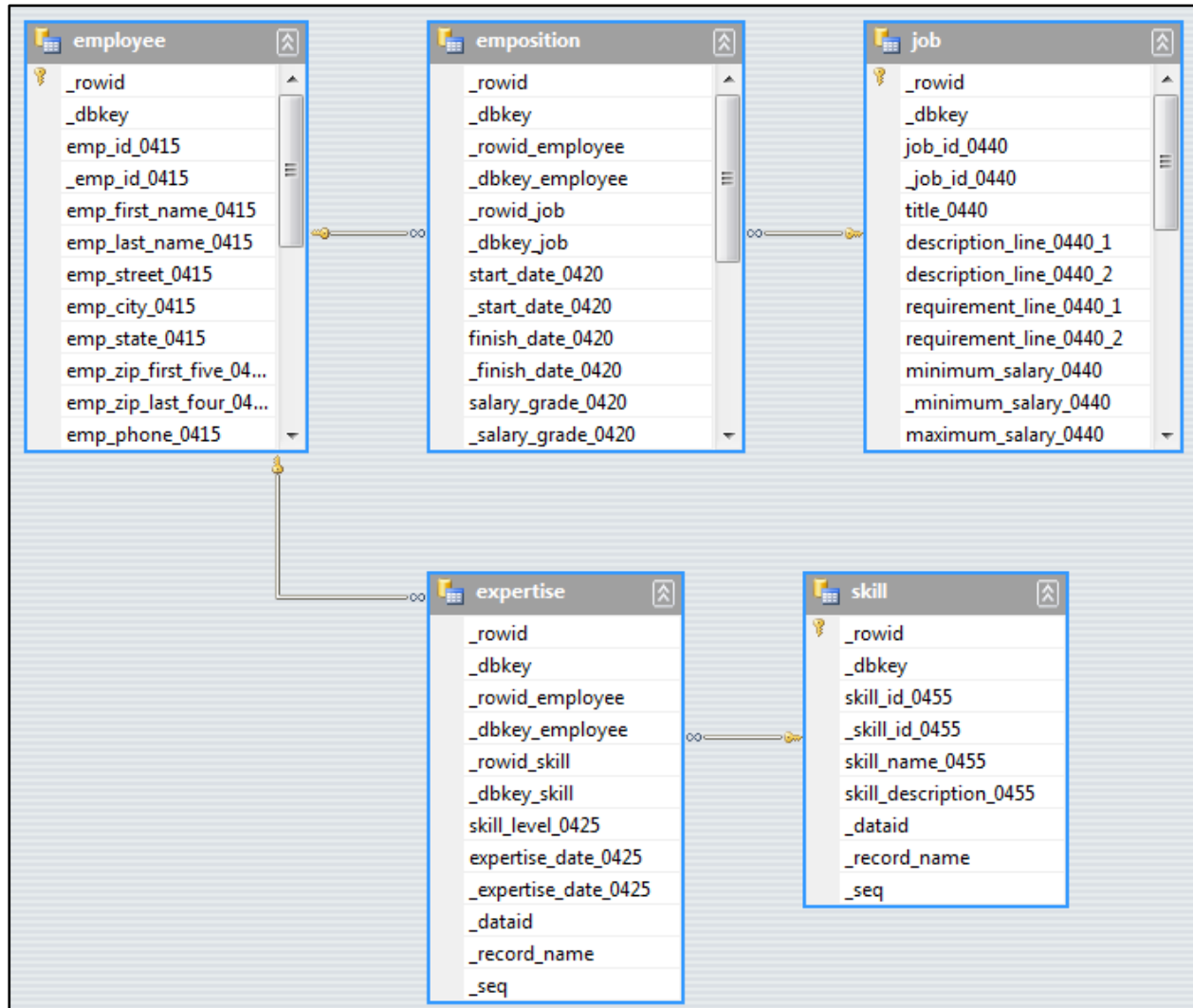
cloud computing

documents rule

- Relational client/server model no longer effective
- Documents rule
- XML allows complex requests and responses
 - SOAP is the protocol of choice
 - SOA is the prevalent design path
- Levels the playing field for CA IDMS

but first, a word about...

XML documents



but first, a word about...

XML documents

```
<XMLQuery_EmployeeInfo>
```

```
<employee _record_name="EMPLOYEE" _seq="1">
```

```
<_rowid>19201793</_rowid>
```

```
<_dbkey>75007:1</_dbkey>
```

```
<emp_id_0415>23</emp_id_0415>
```

```
<_emp_id_0415 xml:space="preserve"> </_emp_id_0415>
```

```
<emp_first_name_0415>Katherine</emp_first_name_0415>
```

```
<emp_last_name_0415>O'Hearn</emp_last_name_0415>
```

```
...
```

```
<_dataid>3C294217</_dataid>
```

```
</employee>
```

but first, a word about...

XML documents

```
<emposition _record_name="EMPOSITION" _seq="73">  
  <_rowid>19203330</_rowid>  
  <_dbkey>75013:2</_dbkey>  
  <_rowid_employee>19201793</_rowid_employee>  
  <_dbkey_employee>75007:1</_dbkey_employee>  
  <_rowid_job>19242759</_rowid_job>  
  <_dbkey_job>75167:7</_dbkey_job>  
  <start_date_0420>1980-12-21T00:00:00-07:00</start_date_0420>  
  <_start_date_0420 />  
  <salary_grade_0420>21</salary_grade_0420>  
  . .  
</emposition>
```

cloud computing requirements

- Multi-platform and heterogeneous
 - XML is the enabler
- Web service consumers don't want to know
 - You must not expose platform-specific nuances
- Document oriented
 - Get it, use it, change it, and store it
 - XML documents are a universal language
- Autonomous transactions
 - You can't serve up relational cursors to your clients in cloud computing
 - Every call either did something or it did nothing

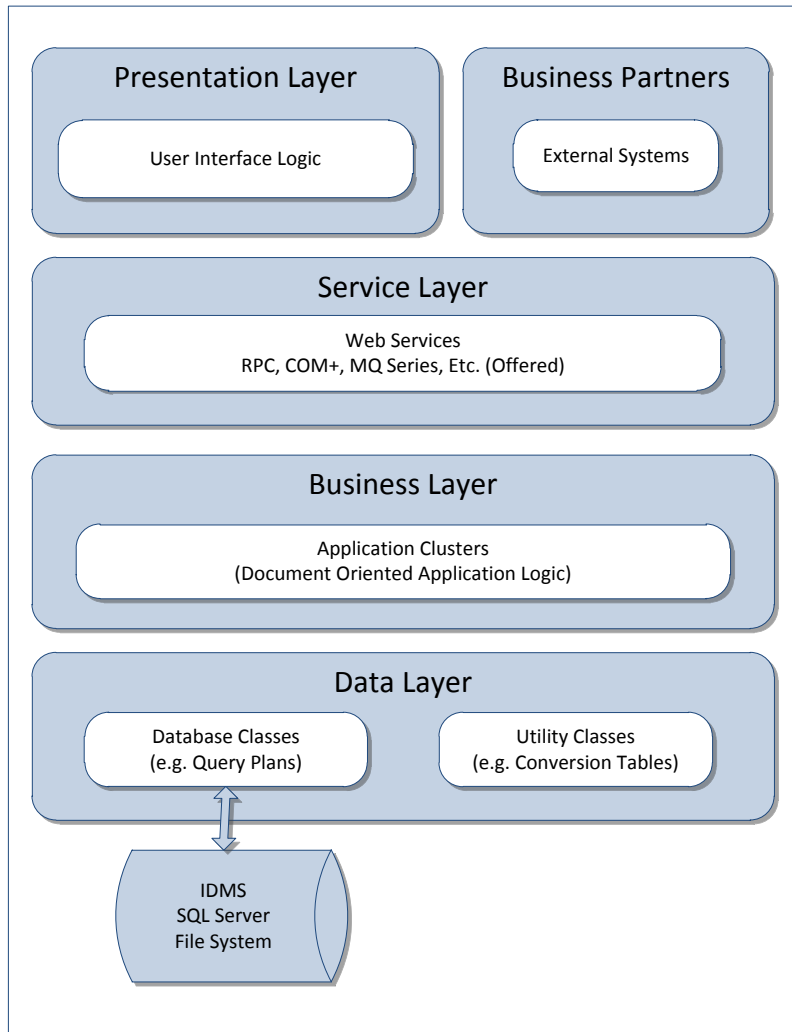
cloud computing requirements

- Scalability
 - Requires linear scalability by adding machines
 - Google seems to do this better than everyone
- Reliability
 - Dropping a server for any reason should not disrupt a user's session
- World-wide performance
 - If the users are world-wide, the server base should be too

*“You wouldn’t build an IDMS
application without
forethought. Cloud computing
is no different.*”

*A well thought out
architecture saves time.”*

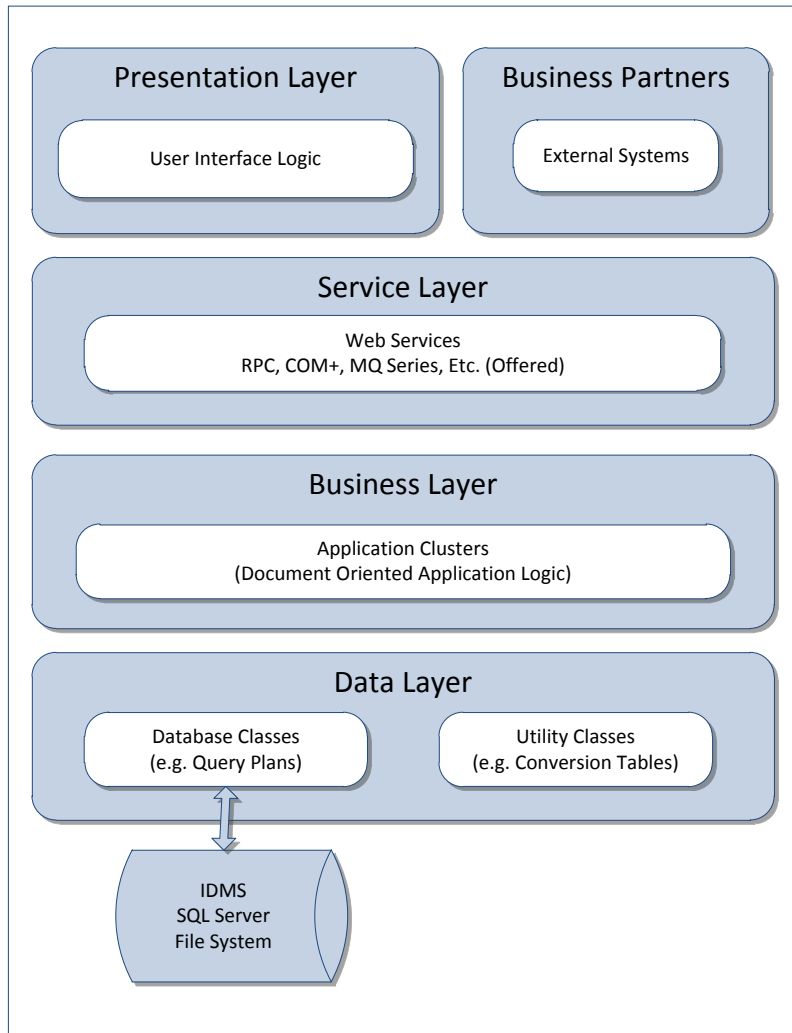
cloud computing software architecture



A layered approach

- Calls flow from the top down
- Rarely skips a level
- Calls NEVER flow up
- Layers are used by referencing DLLs
- Service layers are SOAP web service

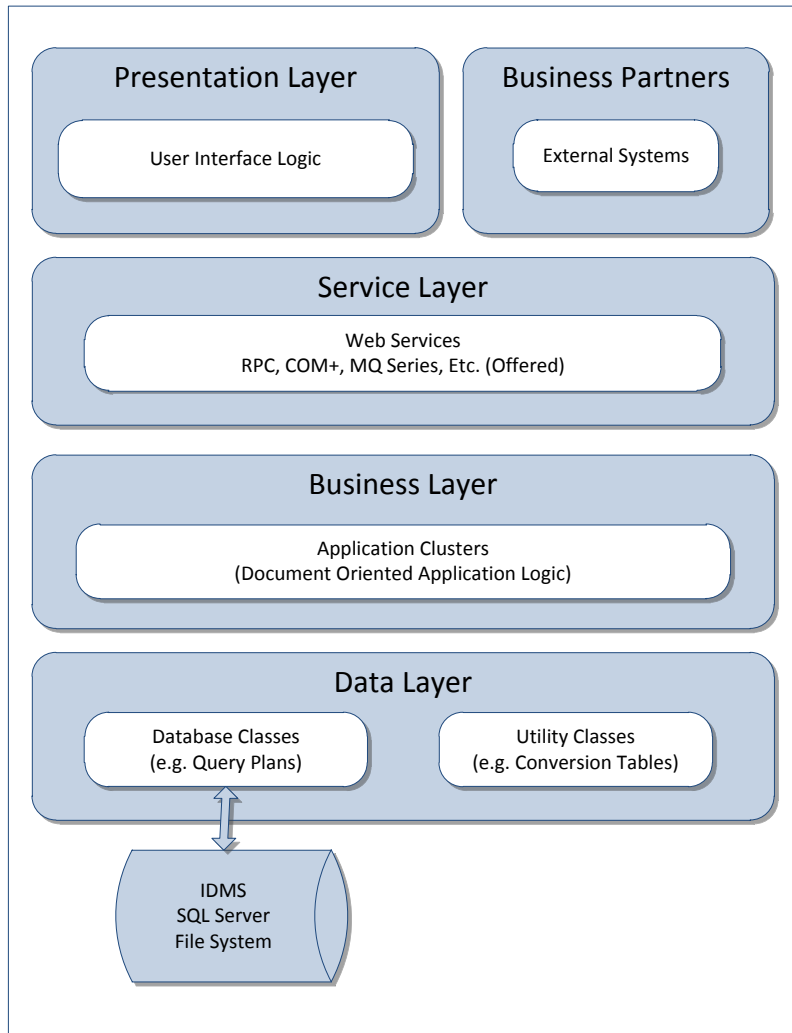
cloud computing software architecture



Data Layer

- Contains all code that directly accesses data
- Can be CA IDMS, relational, files, or calls to outside services like Google Maps
- No business rules here. Merely provides a consistent way to access data by the business layer
- Allows relocation of data without changing higher levels
- CA IDMS data can be supplied by SQL Option or ObjEx XMLQuery

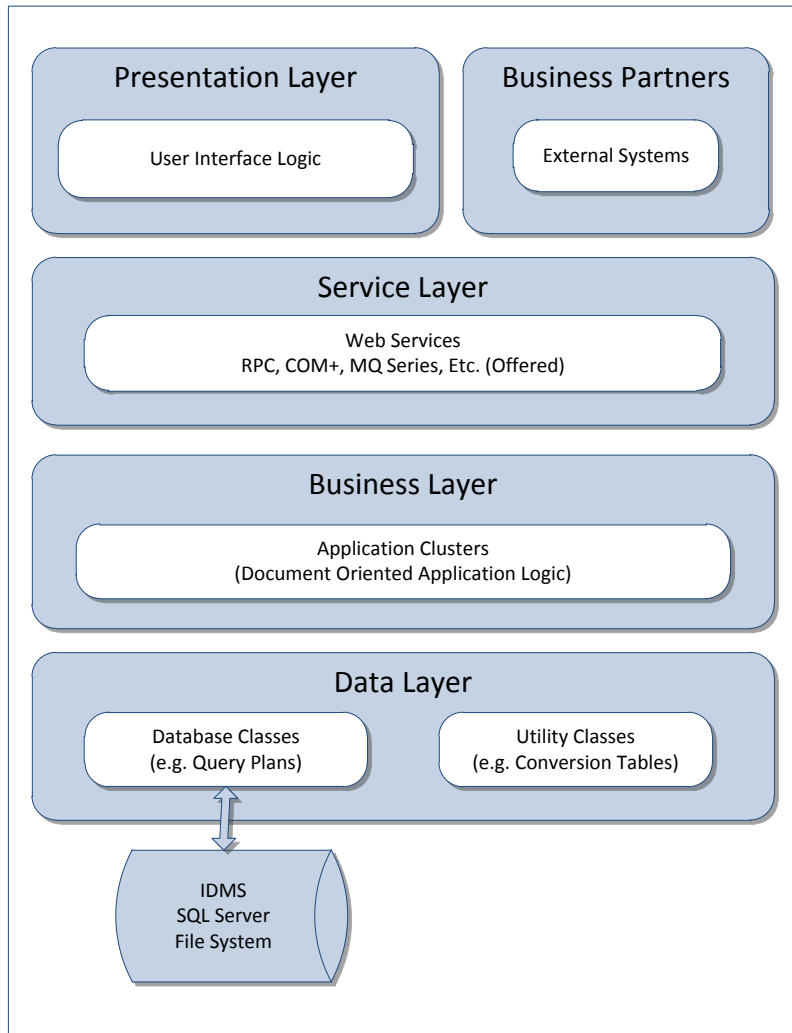
cloud computing software architecture



Business Layer

- Supplies ready-to-use documents
 - Get it, use it, change it, and save it
- Applies all edits and performs transformations
- Some object orientation
 - Could include an ER-like collection of classes
- Document examples
 - Purchase Orders, Sales Orders, others relevant to your organization
 - Reports

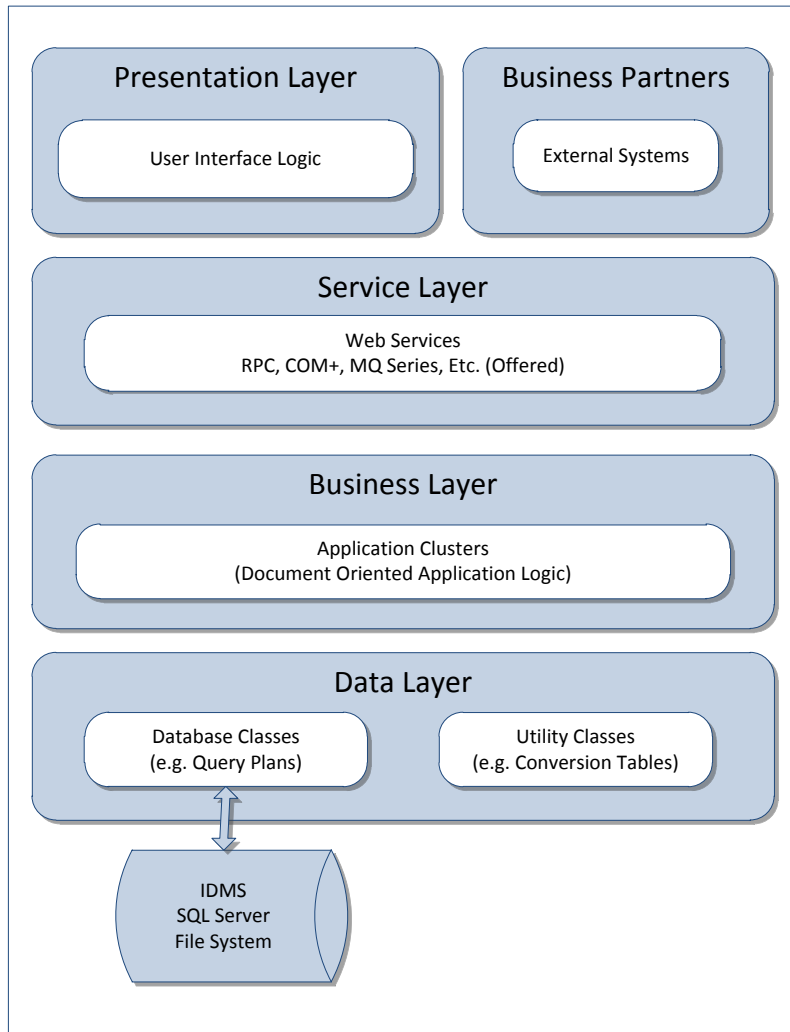
cloud computing software architecture



Service Layer

- Semantically, it simply provides RPC methods that access the Business Layer
- Both private use and for business partners
- Implementation is the key to:
 - Scalability
 - Reliability
 - Performance
- Uses local, system, and global caches
- We'll talk more about caching later on in physical deployment

cloud computing software architecture



Presentation Layer

- The big unknown
 - This layer changes frequently...iPad anyone?
 - Can be windowed, cell phones, PDAs, tablets, mainframe with green screens
- Can be your applications or from your vendors, customers, or even consumers
- Once again, XML rules
- Outside callers always call the Service Layer
- In-house apps may optionally use the Business Layer DLLs directly


cloud computing

nothing's ever perfect – there are challenges

- Rich user interfaces demand data
 - This is why screen scrapers fail in volume situations
- Semi-static data
 - Edit and code tables
 - Why make thousands of calls to CA IDMS?
- Re-fetch
 - Web users tend to surf backward and forward
- Let's talk about data caching
 - Local, machine, cluster

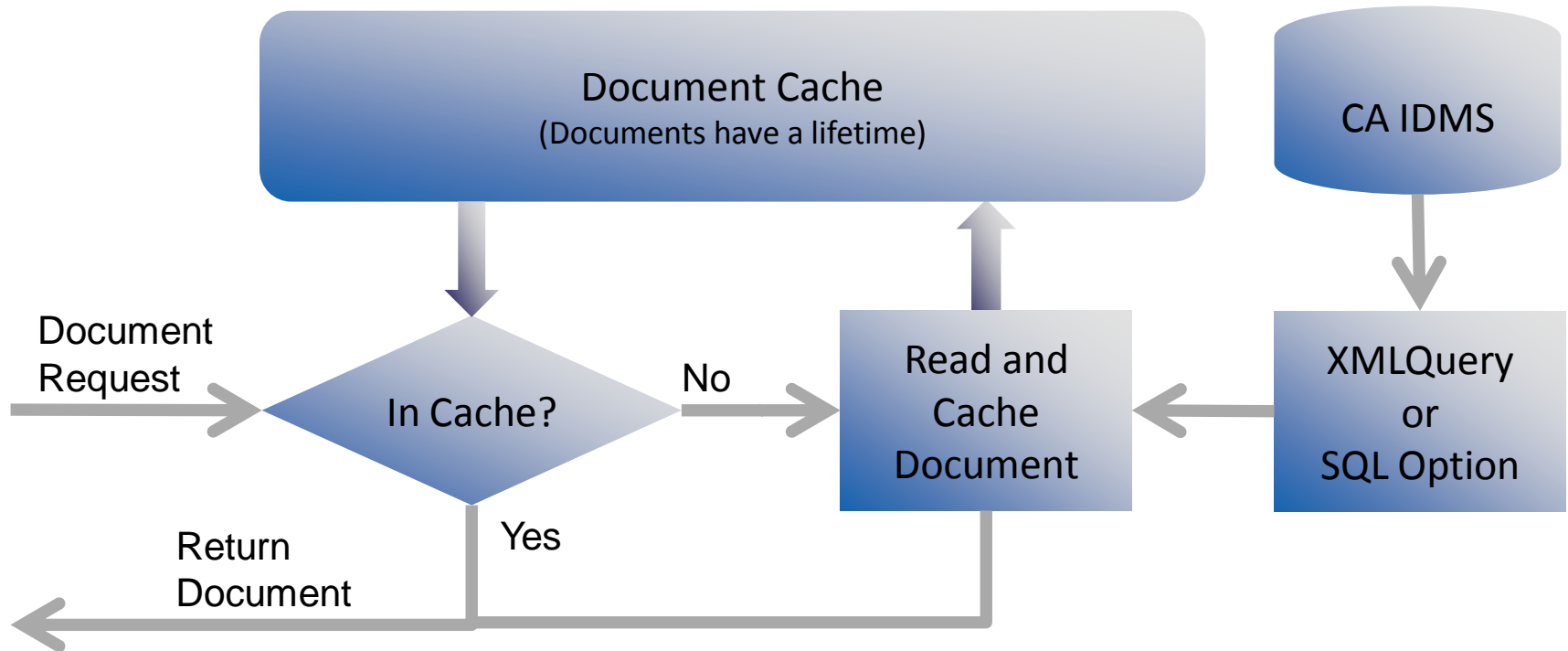
cloud computing

why use a data cache?

- Because the world does not want this: 
- The world wants split-second response times
- And you do not want more hardware
- Caching reduces the demand on your data store

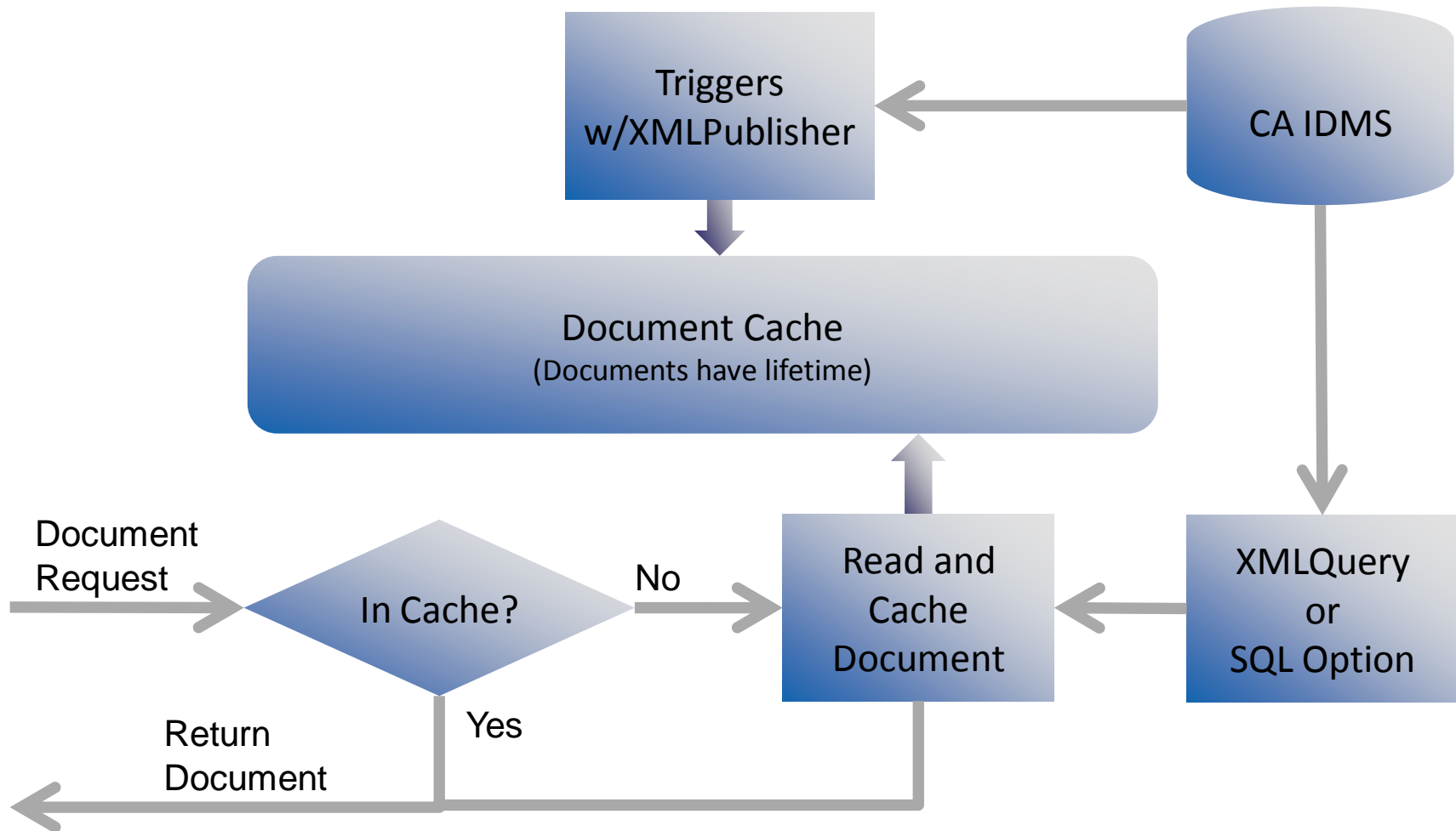
cache usage scenarios

demand driven (pull)



cache usage scenarios

event driven (push and pull)



cache usage

what you might put in

- Data Layer
 - XML documents composed with CA IDMS data
 - Edit and code tables
 - Recent activity from outside web services
- Business Layer
 - Organized documents for reports and forms
 - Information dashboard
- Service Layer
 - User session data
 - Dynamic web page content
 - XML documents transformed
 - Grid
 - Chart
 - Report

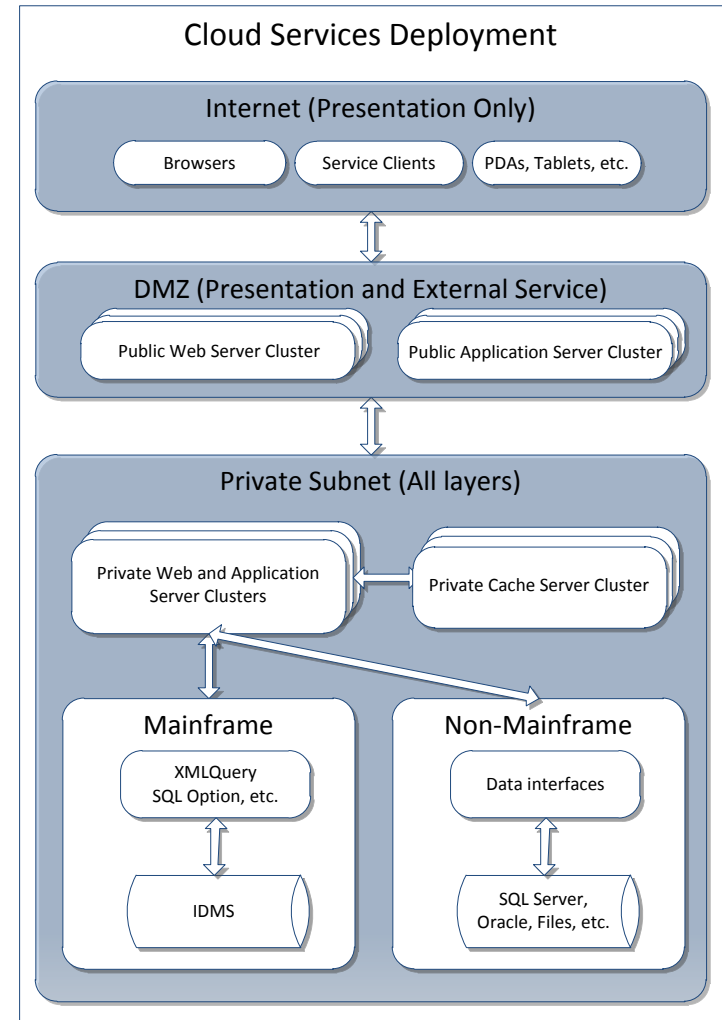
cache usage options

- In-process
 - .NET, Java and C++ all sport in-process classes in their respective foundation class libraries
- Machine-level
 - Products like nCache facilitate caching at the machine level
 - Not reliable as the cache disappears when the machine fails
- Cluster caching
 - Stripes cached data across several machines
 - A little like RAID
 - Scalable and reliable

cloud computing deployment

The key to scalability and reliability

- Internet
 - Presentation only
- DMZ
 - Public web and application server
 - Presentation and Service Layers
- Private Subnet
 - Private network
 - All layers

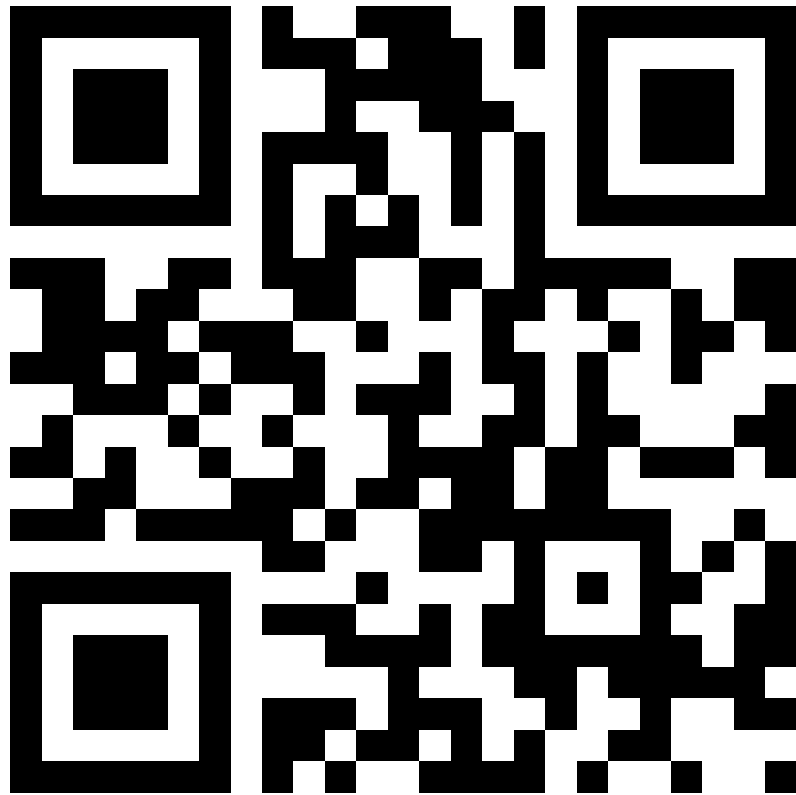


- Cloud computing is a major evolution
 - Document oriented, scalable and reliable
 - Based on architecture and XML
- Cloud computing can be used for both public and private applications
- CA IDMS is a natural for Cloud Computing
 - Use an architecture for success
 - Get it, use it, change it, save it and leverage your current investments

Q&A

ca world[®]'11

Session # MI460SN



MI460SN

**Please scan this
image to fill in
your session
survey on a
mobile device or
complete a hard
copy session
evaluation form**

ca world®'11

thank you

ca world®'11