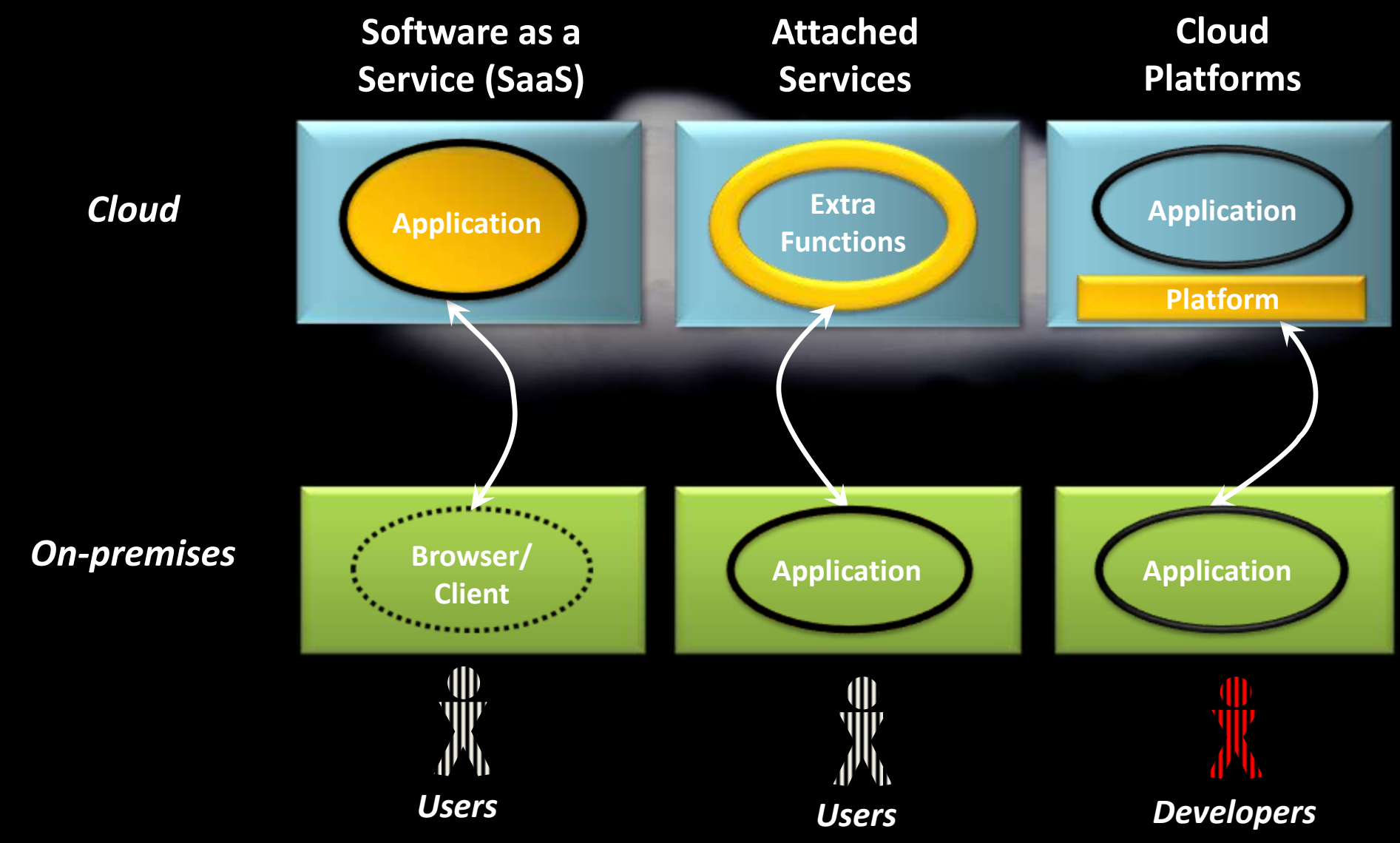
Cloud Platforms Today: A Perspective

David Chappell
Chappell & Associates
April 18, 2009

DavidChappell
www.DavidChappell.com

Cloud Computing

A simple taxonomy



Comparing Cloud Platforms

David Chappell
& Associates

Cloud Platforms
Defining categories

- In the beginning:
  - Infrastructure as a Service (IaaS)
    Example: Amazon Elastic Compute Cloud (EC2)
  - Platform as a Service (PaaS)
    Example: Salesforce.com Force.com

- Today:
  - The cloud platform world is more complex
  - The IaaS/PaaS dichotomy no longer makes much sense

Levels of Functionality

Which is IaaS? Which is PaaS?

	Can run application code	Application runs in a visible VM	Have administrative access to VM	Can create your own OS installation
Amazon Web Services EC2	X	X	X	X
Mosso Cloud Servers	X	X	X	
Windows Azure CTP	X	X		
Google AppEngine	X			
Salesforce.com Force.com	X			

Scenarios
An alternative approach

- A more meaningful way to compare cloud platforms today is by the application scenarios they support

- I've chosen five representative application scenarios
  - There are plenty of other possibilities

Cloud Platform Scenarios (1)

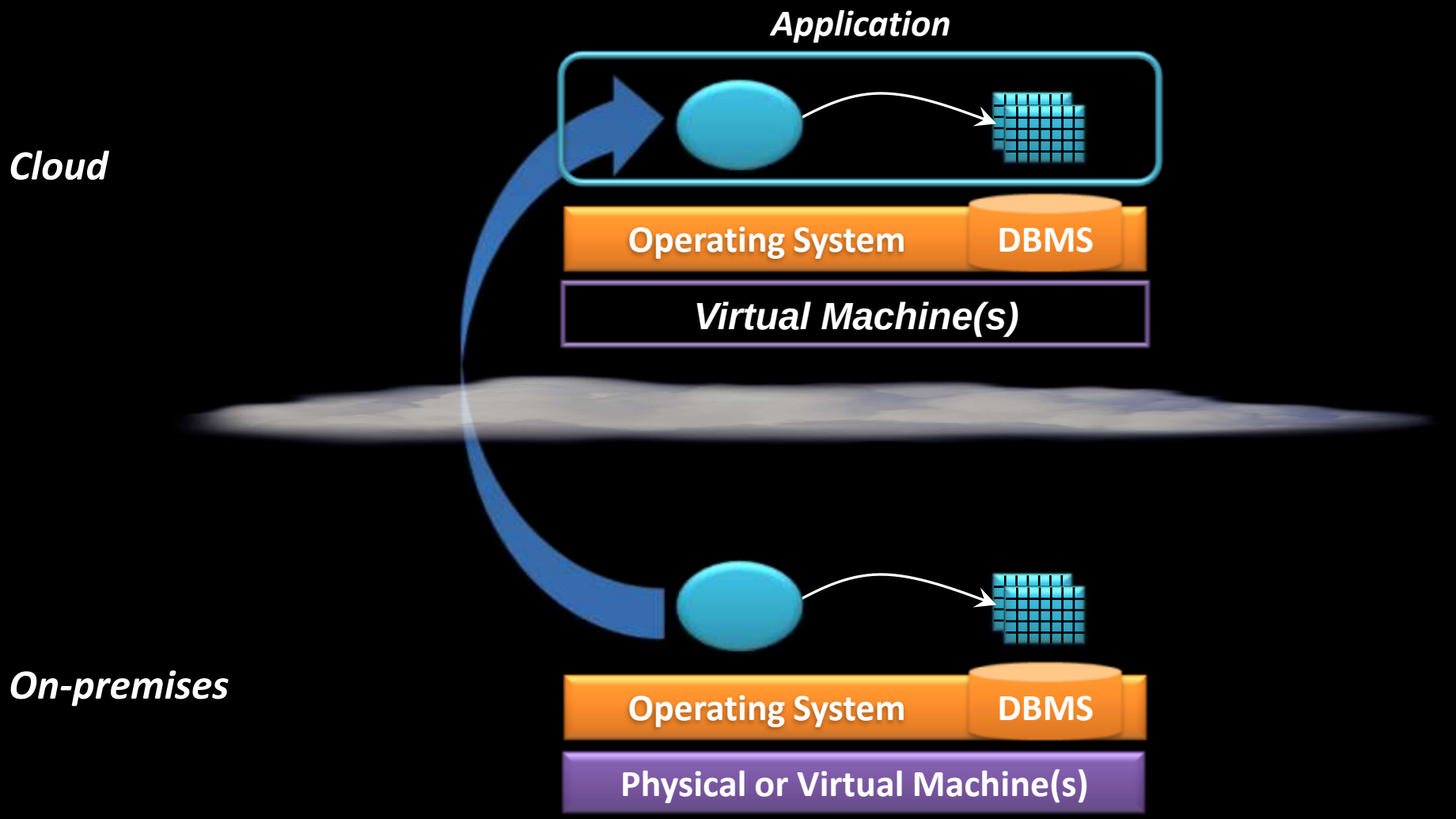
- Running an on-premises app unchanged in the cloud
  - Example: An enterprise moving an on-premises line-of-business app to the cloud
- Creating a moderately scalable Web app
  - Example: An enterprise building a custom application used by its employees
  - Example: An ISV creating a moderately popular consumer application

Cloud Platform Scenarios (2)

- Creating a very scalable Web application
  - Example: An ISV creating the next Facebook
  - Example: A large bank creating a customer-facing app
- Creating a parallel processing application
  - Example: An enterprise using MapReduce to process large amounts of data
- Creating a very scalable Web application with background processing
  - Example: An ISV creating the next YouTube

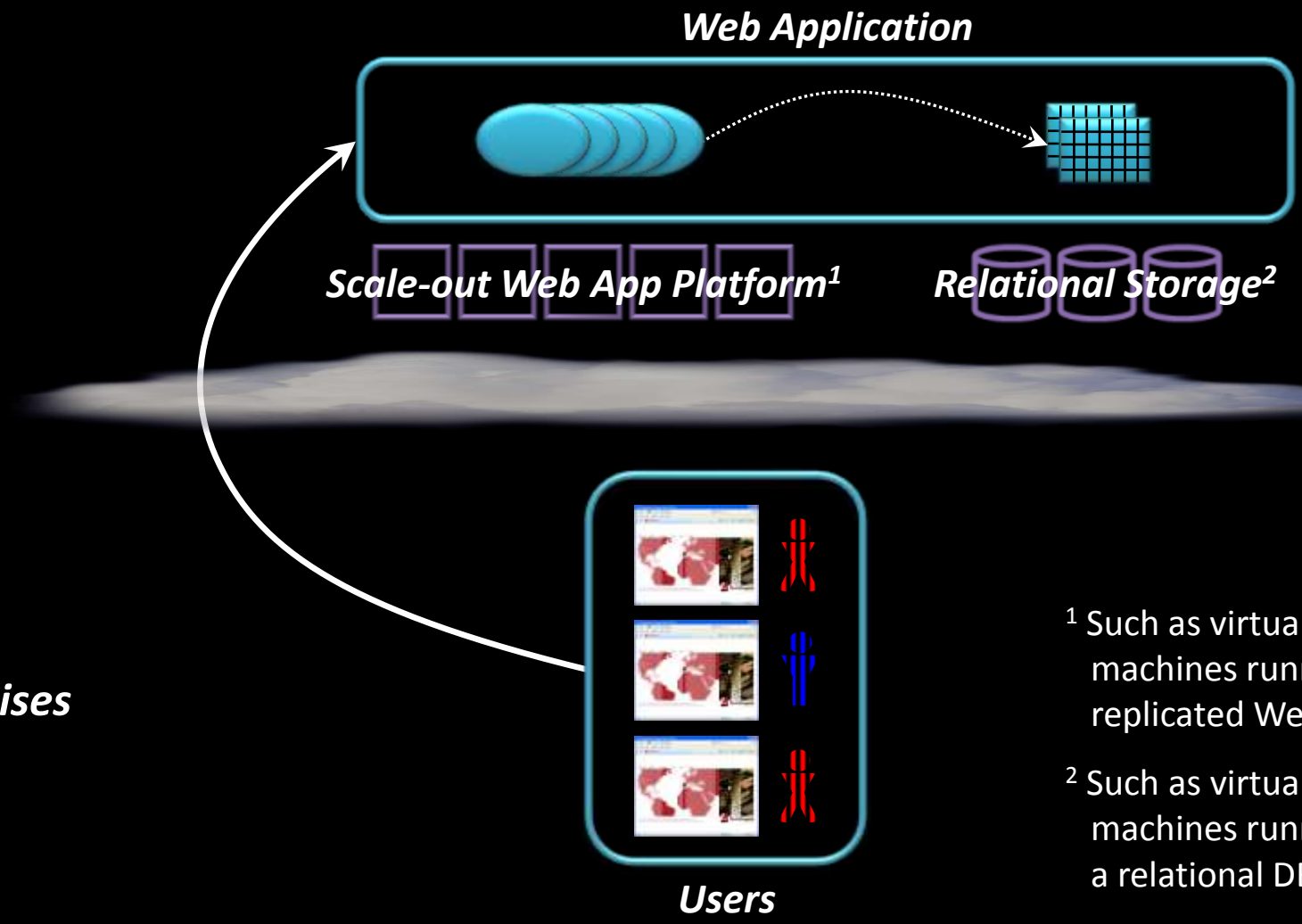
Cloud Platform Scenarios

Running an on-premises application unchanged



Cloud Platform Scenarios

Creating a moderately scalable Web application

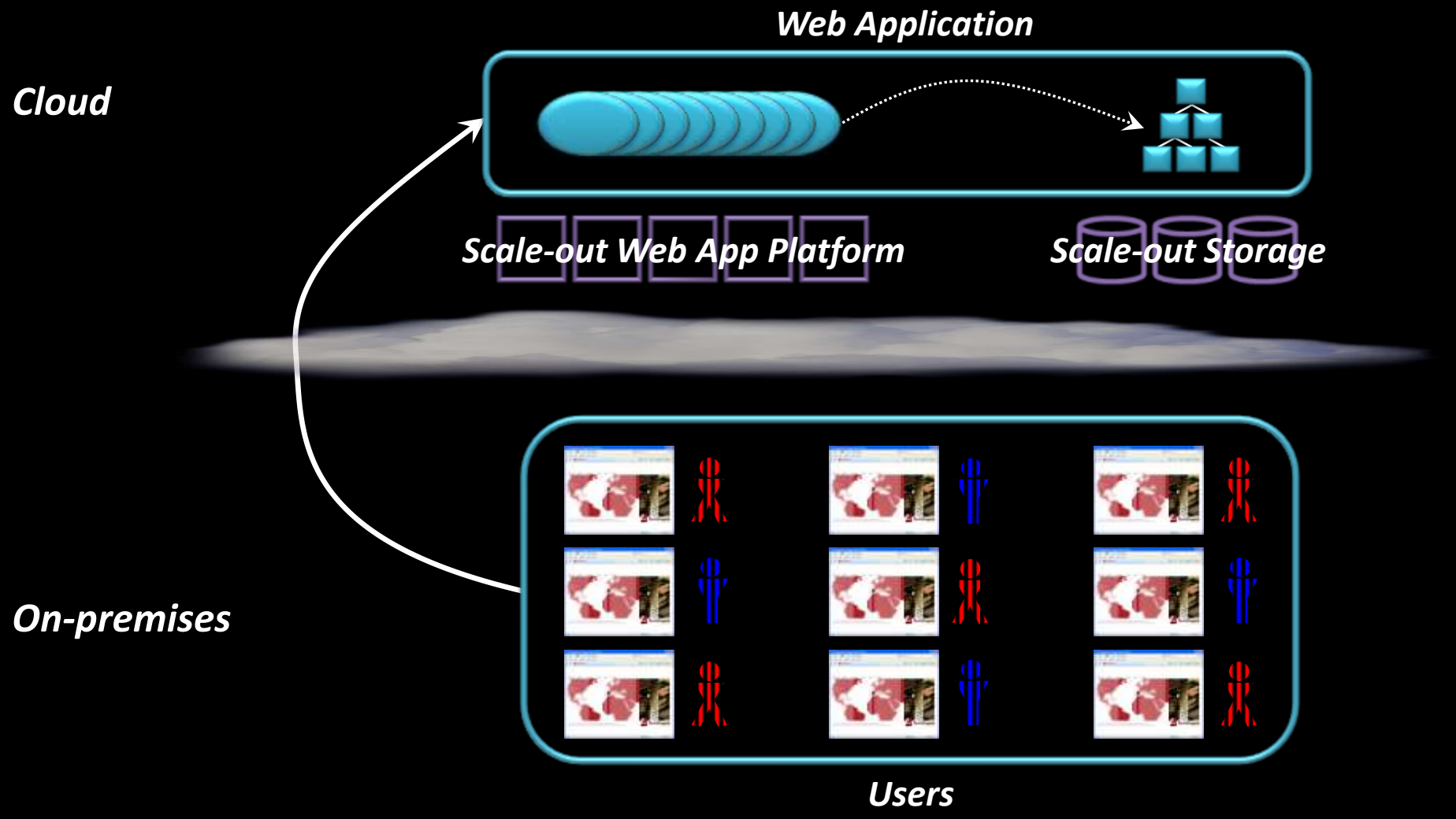


[1] Such as virtual machines running replicated Web servers

[2] Such as virtual machines running a relational DBMS

Cloud Platform Scenarios

Creating a very scalable Web application

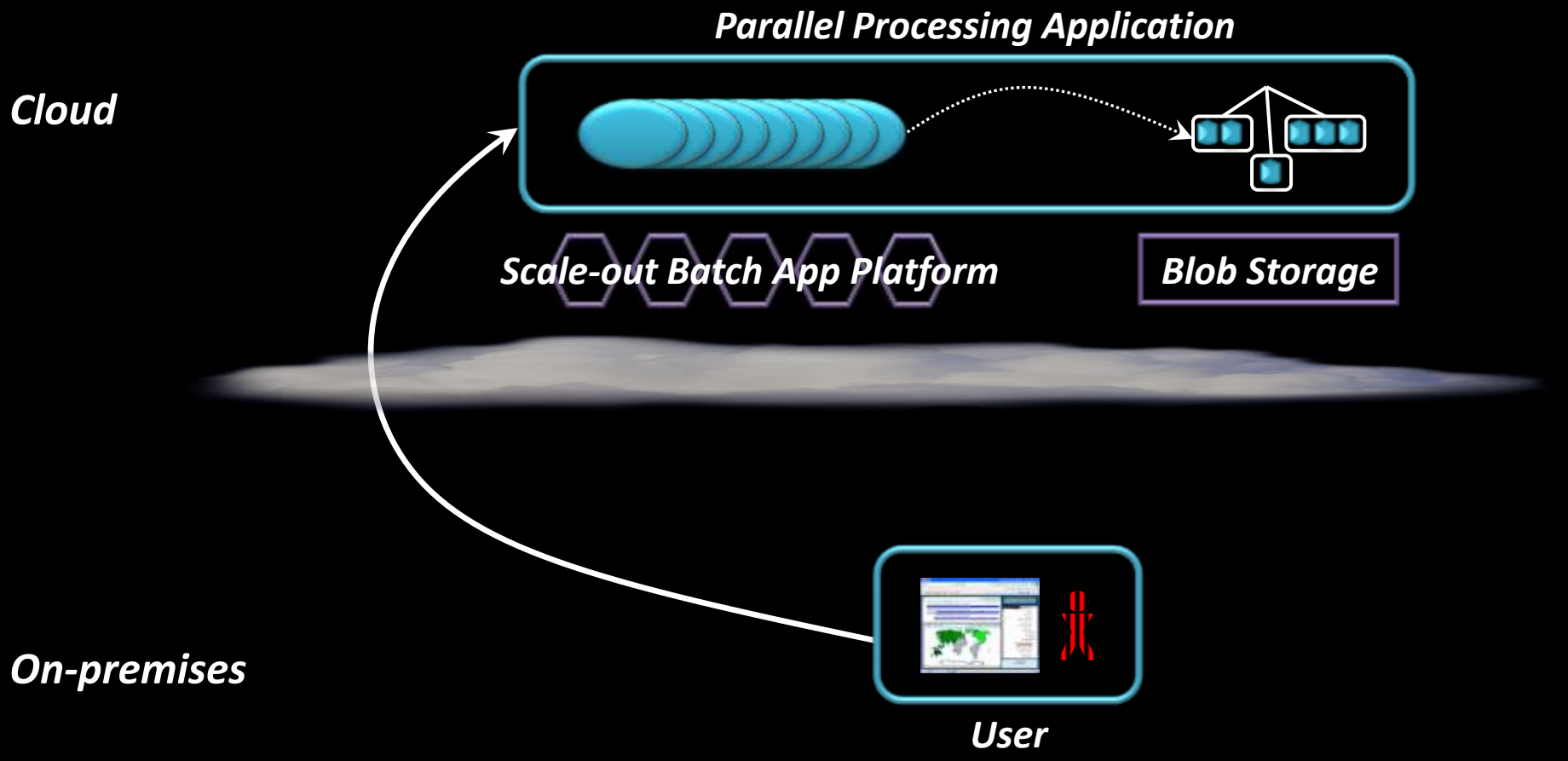


Examining Scale-Out Storage
An aside

- Challenges:
  - An unfamiliar structure
    - Moving relational data to it takes work
    - Supporting services are scarce, e.g., reporting
  - No SQL
  - No schema
  - Some platform lock-in
    - Because each vendor does it differently
- Benefits
  - Massive scalability

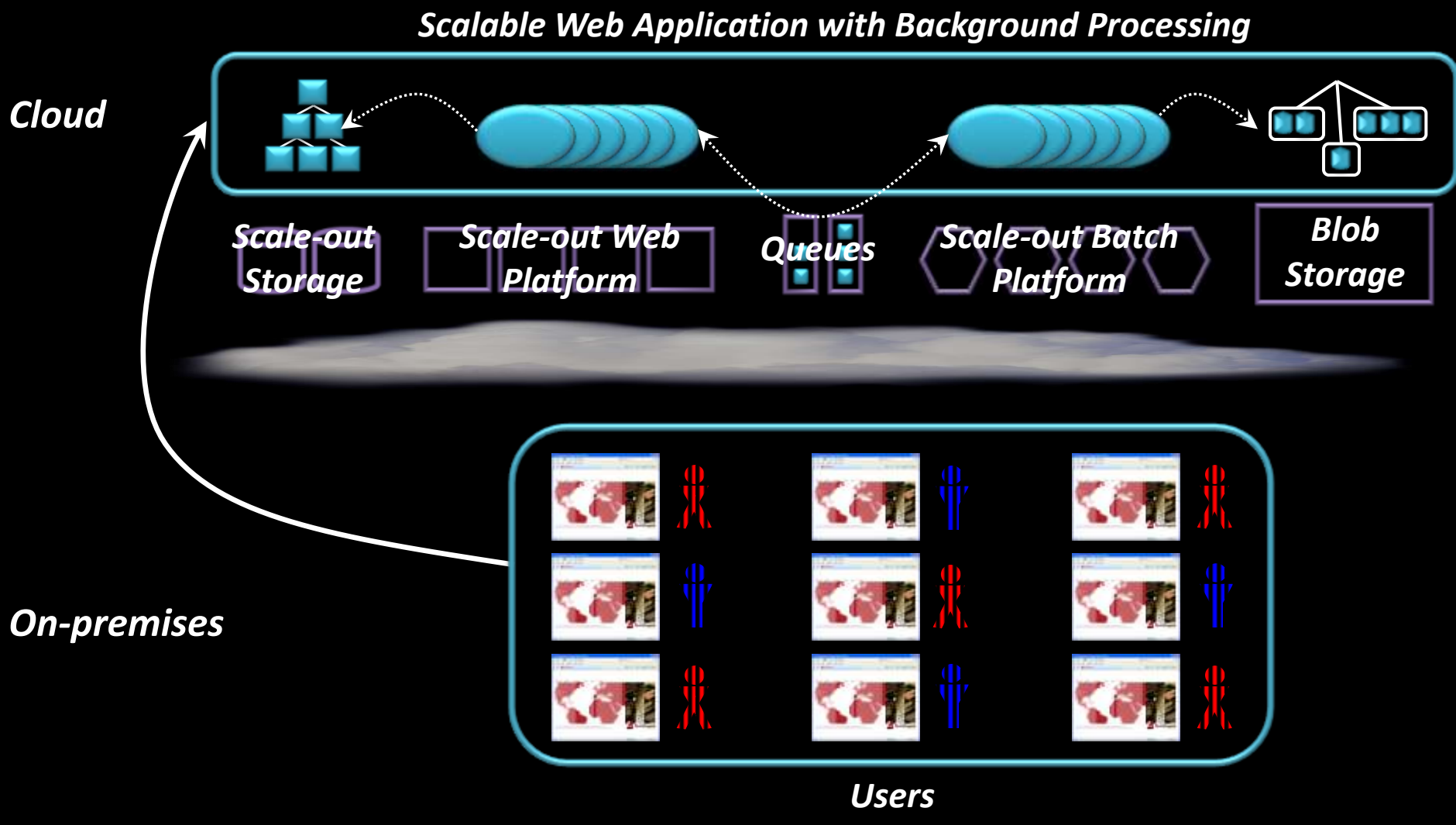
Cloud Platform Scenarios

Creating a parallel processing application



Cloud Platform Scenarios

A scalable Web app with background processing



Scenarios vs. Platform Capabilities

	Run On-Premises Apps	Create Moderately Scalable Web Apps	Create Very Scalable Web Apps	Create Parallel Processing Apps	Create Very Scalable Web Apps with Background Processing
VM with standard OS	X				
Scale-out web app platform		X	X		X
Scale-out batch app platform				X	X
Relational storage	X	X			
Scale-out storage			X		X
Blob storage				X	X
Queues					X

GoGrid, Mosso, Flexiscale, Others

Typical scenarios

	Run On-Premises Apps	Create Moderately Scalable Web Apps	Create Very Scalable Web Apps	Create Parallel Processing Apps	Create Very Scalable Web Apps with Background Processing
VM with standard OS	*VMs*				
Scale-out web app platform		*VMs*			
Scale-out batch app platform					
Relational storage	*VMs (w/RDBMS)*	*VMs (w/RDBMS)*			
Scale-out storage					
Blob storage					
Queues					

Amazon Web Services

Typical scenarios

	Run On-Premises Apps	Create Moderately Scalable Web Apps	Create Very Scalable Web Apps	Create Parallel Processing Apps	Create Very Scalable Web Apps with Background Processing
VM with standard OS	EC2 VMs				
Scale-out web app platform		EC2 VMs	EC2 VMs		EC2 VMs
Scale-out batch app platform				EC2 VMs , Elastic MapReduce	EC2 VMs
Relational storage	EC2 VMs (w/RDBMS)	EC2 VMs (w/RDBMS)			
Scale-out storage			SimpleDB		SimpleDB
Blob storage				Simple Storage Service (S3)	S3
Queues					Simple Queue Service (SQS)

Windows Azure CTP
Typical scenarios

	Run On-Premises Apps	Create Moderately Scalable Web Apps	Create Very Scalable Web Apps	Create Parallel Processing Apps	Create Very Scalable Web Apps with Background Processing
VM with standard OS					
Scale-out web app platform		*Web role*	*Web role*		*Web role*
Scale-out batch app platform				*Worker role*	*Worker role*
Relational storage		*SQL Data Services*			
Scale-out storage			*Tables*		*Tables*
Blob storage				*Blobs*	*Blobs*
Queues					*Queues*

SQL Data Services (SDS)
An aside

- Provides relational storage in the cloud
  - A distinct service (not part of Windows Azure)
  - Built on Microsoft SQL Server technology
- Can be used by applications running in the cloud or on-premises

Google AppEngine
Typical scenarios

	Run On-Premises Apps	Create Moderately Scalable Web Apps	Create Very Scalable Web Apps	Create Parallel Processing Apps	Create Very Scalable Web Apps with Background Processing
VM with standard OS					
Scale-out web app platform			*Java/Python runtime*		
Scale-out batch app platform					
Relational storage					
Scale-out storage			*Datastore*		
Blob storage					
Queues					

Salesforce.com Force.com

Typical scenarios

	Run On-Premises Apps	Create Moderately Scalable Web Apps	Create Very Scalable Web Apps	Create Parallel Processing Apps	Create Very Scalable Web Apps with Background Processing
VM with standard OS					
Scale-out web app platform			*Force.com runtime*		
Scale-out batch app platform					
Relational storage					
Scale-out storage			*Force.com storage*		
Blob storage					
Queues					

Comparing AppEngine and Force.com
Two different approaches

- AppEngine provides a Java or Python environment for building Web apps
  - Apps are built by developers
- Force.com provides a platform for creating data-oriented business applications with browser interfaces
  - Apps can be built by less technical people
  - Apps can also be built by developers
    - Using Apex Code

Comparing Cloud Platforms

Summarizing typical scenarios

	Run On-Premises Apps	Create Moderately Scalable Web Apps	Create Very Scalable Web Apps	Create Parallel Processing Apps	Create Very Scalable Web Apps with Background Processing
GoGrid, Mosso, Flexiscale, etc.	X	X			
Amazon Web Services	X	X	X	X	X
Windows Azure CTP		X	X	X	X
Google AppEngine			X		
Salesforce.com Force.com			X		

Cloud Platforms Today:
Other Topics

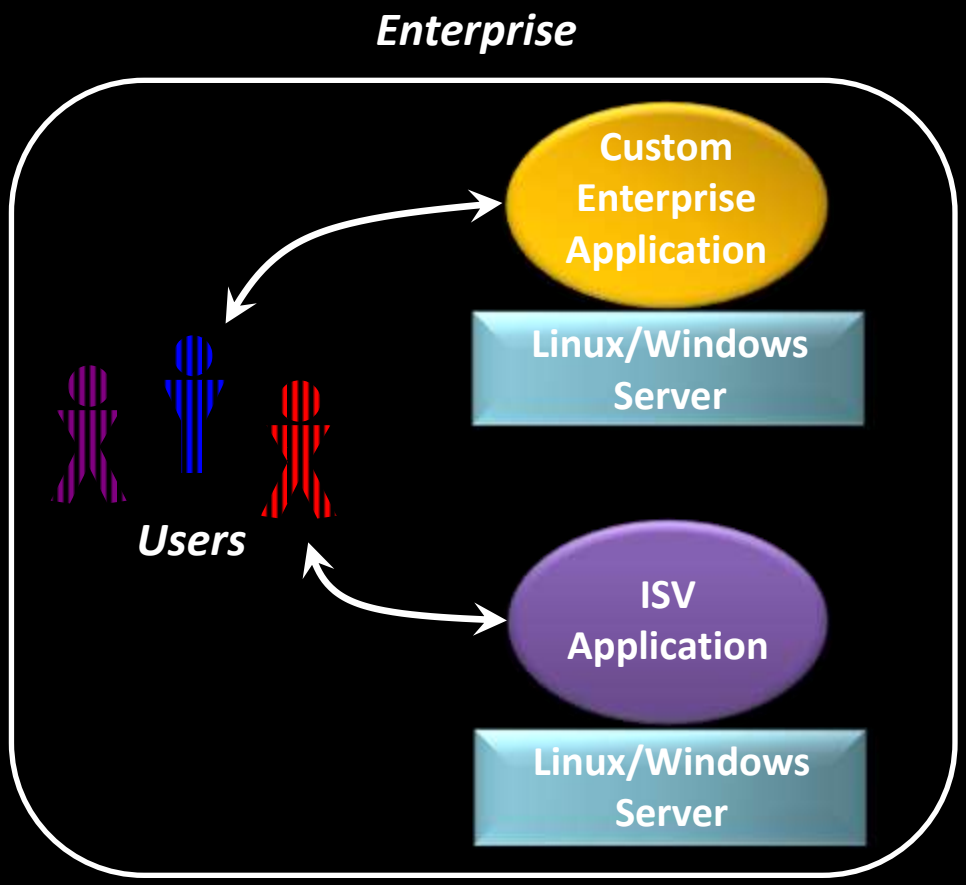
David Chappell
& Associates

Categorizing the Scenarios
Who cares most about each one?

- Enterprises
  - Run an on-premises app unchanged
  - Others of interest:
    - Create moderately scalable Web apps
    - Create parallel processing apps
- ISVs
  - Create very scalable Web apps
  - Create very scalable Web apps with background processing

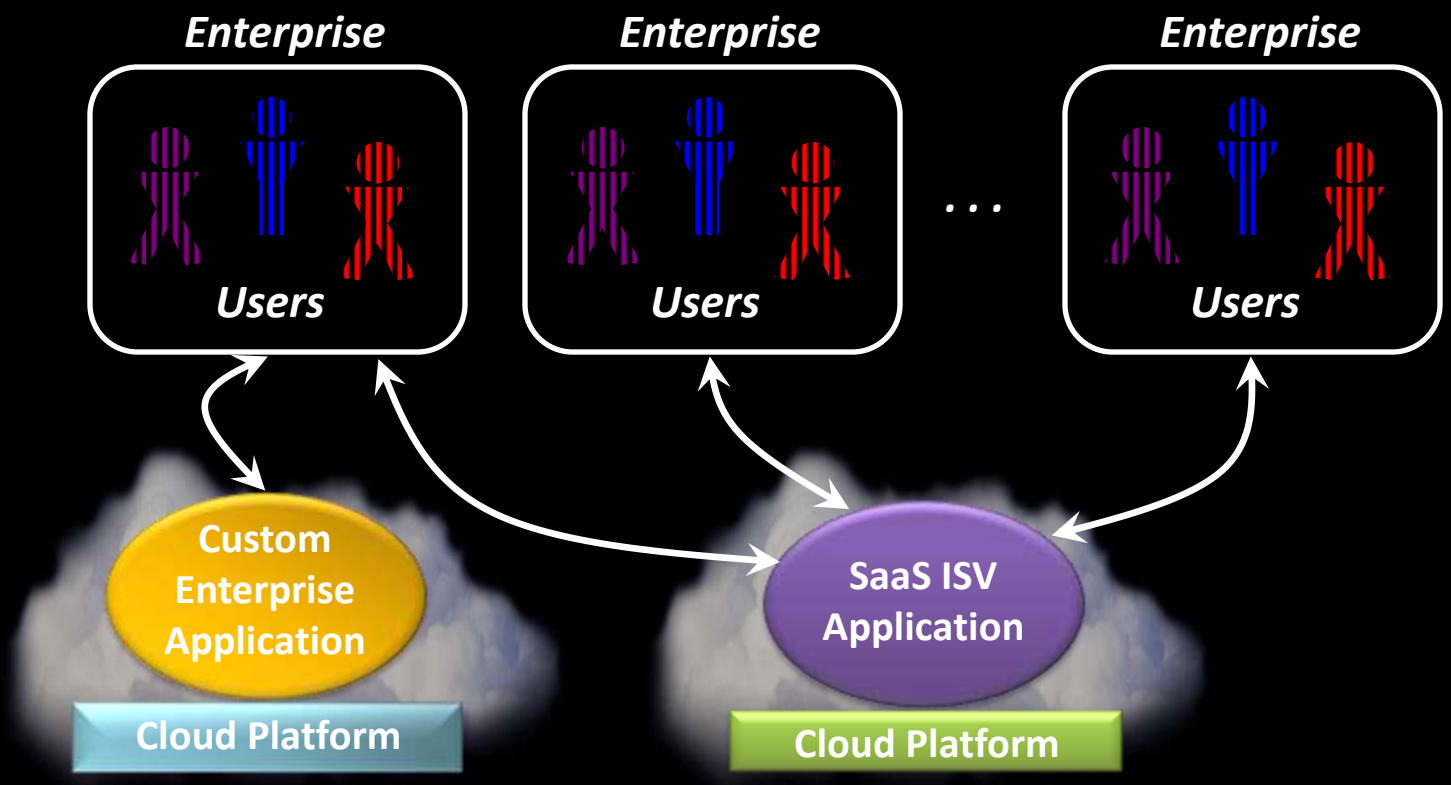
The World Before SaaS Applications

- ISV apps and custom enterprise apps have similar scalability/reliability requirements



The World with SaaS Applications

- ISV apps and custom enterprise apps have different scalability/reliability requirements



Cloud Platform Bifurcation?

- Enterprises
  - Want cloud platforms to lower their costs
  - Might want to be able to move apps and data easily between the cloud and on-premises
- ISVs
  - Want cloud platforms to provide scalability and reliability

- These different requirements are met by different cloud platform capabilities

Pricing
Some examples

- Amazon Web Services
  - Pay for your data storage and transfer
  - Pay per-hour for each of your running VMs
- Microsoft Windows Azure
  - Not yet announced
- Google AppEngine
  - Pay for your data storage and transfer
  - Pay for the CPU time your app uses
- Salesforce.com Force.com
  - Pay per user or based on page views

Pricing
The importance of reservations

- EC2 now offers reservations
  - This lowers prices for longer-term commitments
- Reservations help a cloud platform vendor do better capacity planning
  - And thus make more money
- Expect the idea to spread

Cloud Platform Lock-in
Interoperability

- Questions:
  - How easy is it to expose my cloud platform app's *services* to other apps?
  - How easy is it to access my cloud platform app's *data* from other apps?
- The situation today is quite good:
  - Cloud platform apps can expose and consume Web services
  - Cloud platform app data can be accessed by external apps

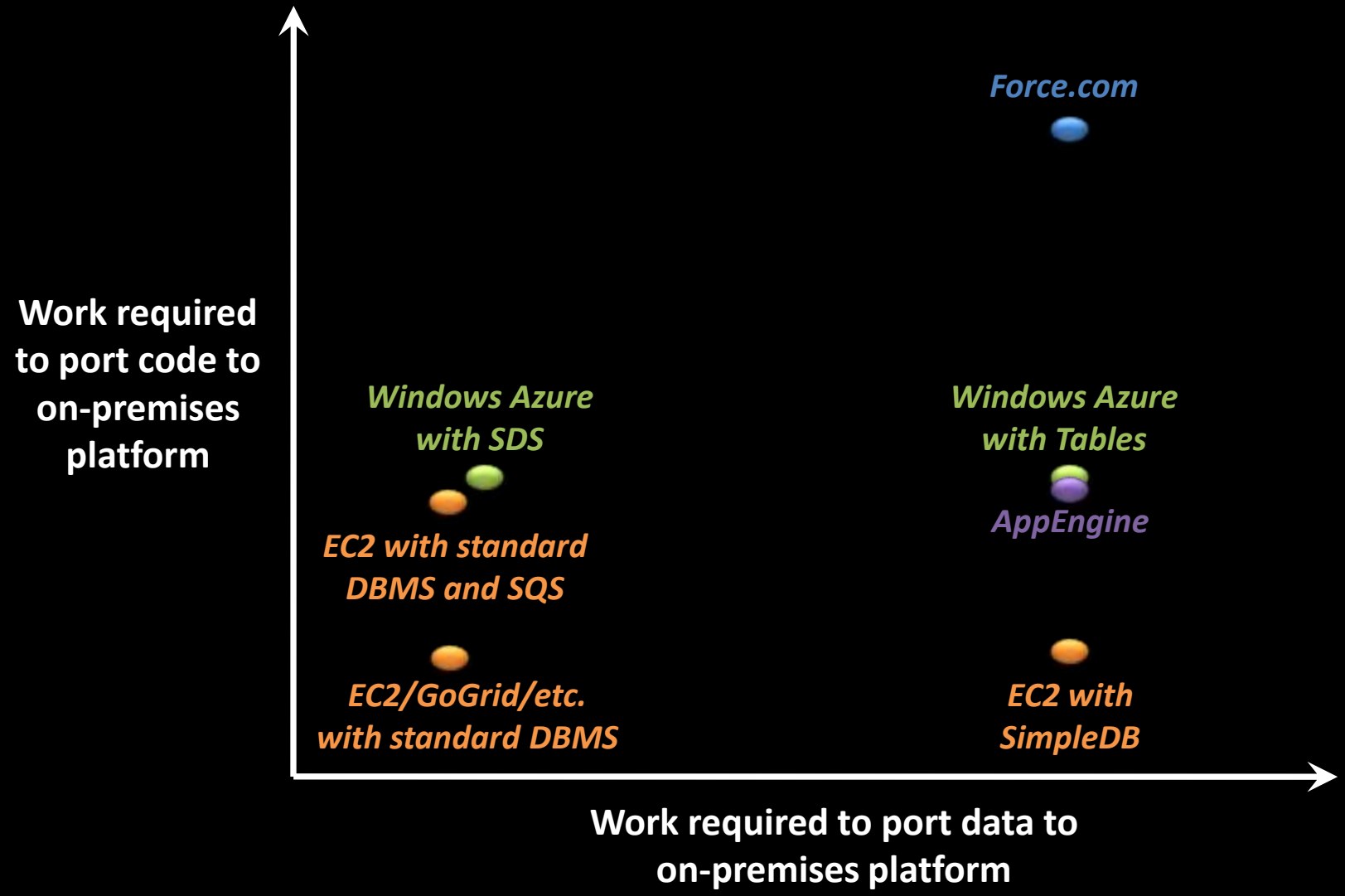
Cloud Platform Lock-in
Portability

- Questions:
  - How easy is it to move my cloud platform *code* back on-premises or to another cloud platform?
  - How easy is it to move my cloud platform *data* back on-premises or to another cloud platform?

- The situation today varies with different cloud platforms
  - And different options within those platforms

Cloud Platform Lock-in

Estimating relative portability for code and data



Conclusion

- Today's cloud platform choices vary considerably
  - It's more than just IaaS and PaaS
- Making the right decision requires:
  - Knowing what your goals are
  - Understanding what each option offers
- It's a great time to build applications!

About the Speaker

David Chappell is Principal of Chappell & Associates (www.davidchappell.com) in San Francisco, California. Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better decisions about new technology. David has been the keynote speaker for many events and conferences on five continents, and his seminars have been attended by tens of thousands of IT decision makers, architects, and developers in forty countries. His books have been published in a dozen languages and used regularly in courses at MIT, ETH Zurich, and other universities. In his consulting practice, he has helped clients such as Hewlett-Packard, IBM, Microsoft, Stanford University, and Target Corporation adopt new technologies, market new products, train their sales staffs, and create business plans. Earlier in his career, David wrote networking software, chaired a U.S. national standards working group, and played keyboards with the Peabody-award-winning Children's Radio Theater. He holds a B.S. in Economics and an M.S. in Computer Science, both from the University of Wisconsin-Madison.

www.davidchappell.com