PDFTRON

PDF vs Office

Is Reliable Office Rendering Possible?

Motivation and Overview

- Office is a good format for document creation
- PDF is better for sharing, viewing and archival
  - Will render similarly on different devices
  - Will render similarly now and in the future
  - Makes better use of limited hardware
- Can create workflows that get the best out of both

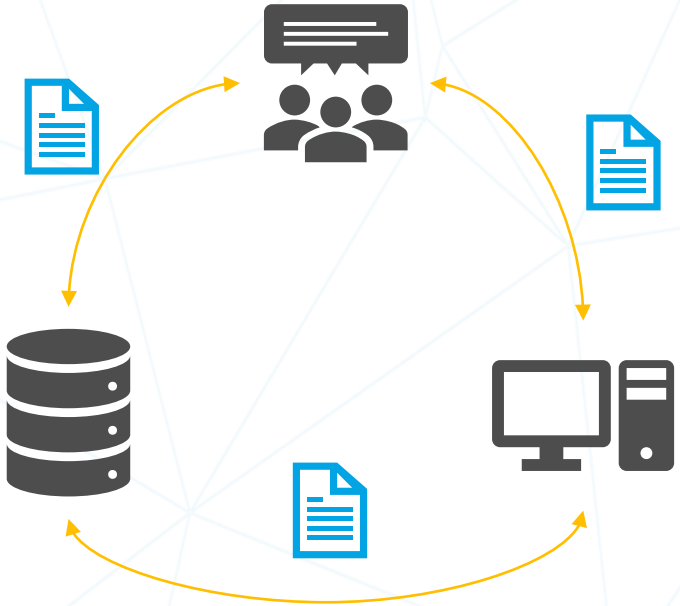
What is an Office file?

- A document in the Office Open XML (OOXML) format
- Standardized as ECMA-376, and later as ISO/IEC 29500
- Typically produced by Microsoft Office

Roadmap

- Introduction
- Next: Defining the problem
  - Viewing, sharing, archival, editing
- An in depth look
  -
  -
  -
  -
- Solutions
  -
  -
  -
- Conclusion

Office Files Are Important



Internal Processes

Directly from Users

Roadmap

- Introduction
- Defining the problem
  - Viewing, sharing, archival, editing
- Next: An in depth look
  - PDF for viewing, Office for editing
  -
  -
  -
- Solutions
  -
  -
  -
- Conclusion

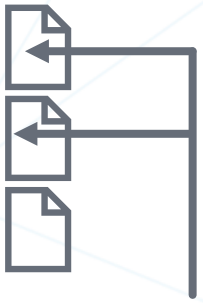
Now For the Main Event!

electronic document CONFERENCE

Compare and Contrast – Layout

PDF	Office
Fixed: • All elements in a specific location 	**Reflowable:** • Elements placed in order • Fill available space 

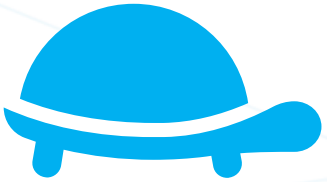
Compare and Contrast – File Structure

PDF	Office
• Flat file with look-up table • Fast per-page access 	• Zip with XML and resources • Not split by page*  <XML>

*Mostly: presentationML documents *are* split by page

A Case for PDF: Interactive Viewing

- Office layout is CPU heavy
- PDF offers faster, more predictable viewing
  - Important for web and mobile environments.



A Case for Office: Content Creation

- Reflow is important for document editing
  - Especially true for word processing!
- Office formats are a good fit for content creation/editing

Roadmap

- Introduction
- Defining the problem
  - Viewing, sharing, archival, editing
- Next: An in depth look
  - PDF for viewing, Office for editing
  - Office rendering is unreliable
  -
  -
- Solutions
  -
  -
  -
- Conclusion

Compare and Contrast – Fonts

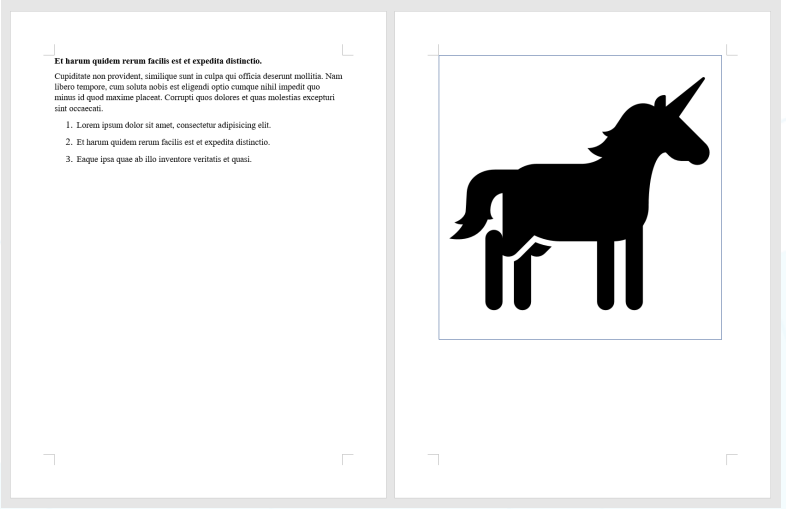
PDF	Office
• Mostly embedded • All fonts retain widths 	• Mostly not embedded • No size information 

Communicating with others

- Everyone needs to see the same thing
- Sometimes this is legally required
- Reflowable layout makes this tough

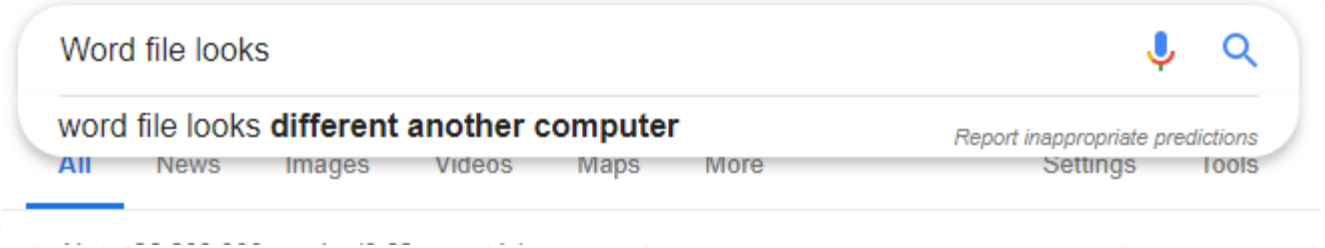
Alice creates a document:

Bob views the document:



A Second Opinion

Google search prediction tells us something similar:



Roadmap

- Introduction
- Defining the problem
  - Viewing, sharing, archival, editing
- Next: An in depth look
  - PDF for viewing, Office for editing
  - Office rendering is unreliable
  - Office rendering will never be reliable
  - PDF for sharing and archival!
- Solutions
  -
  -
  -
- Conclusion

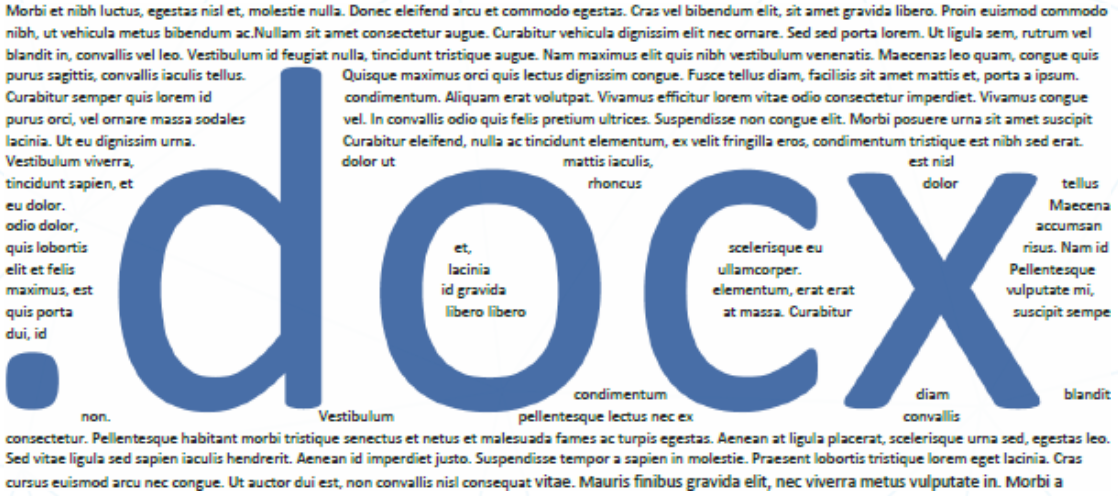
Compare and Contrast – Specification

PDF	Office
• 748 pages (1.7) • 950 pages (2.0) • Complete – could feasibly build a working renderer from the spec alone.	• ~8800 pages (ECMA 376 4th ed.) • Woefully incomplete. • Render behaviour must be reverse-engineering.

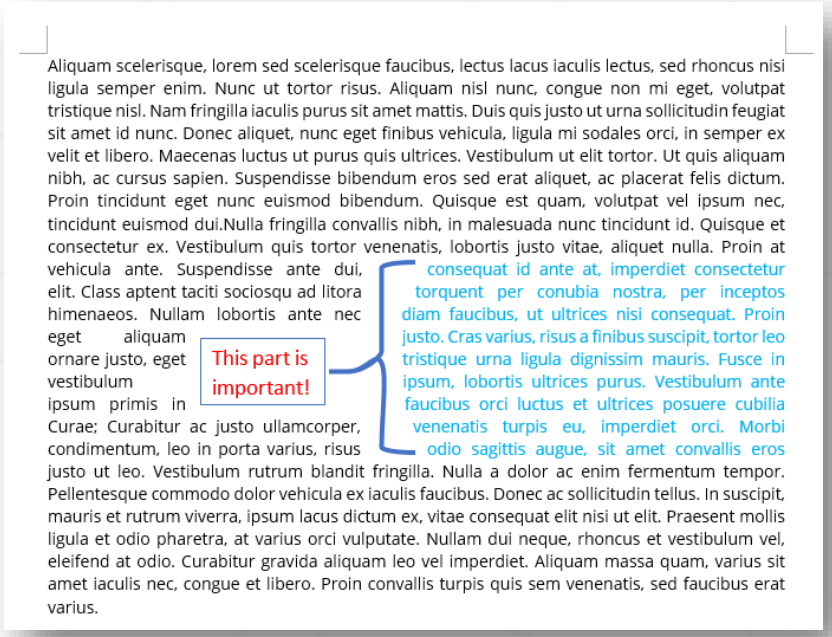
Technical Challenges – Floating Content

OOXML has a very feature-rich floating content model:

- Text can wrap on either or both sides
- Floats can be anchored to any previous inline content
- Much more complex than analogous html feature

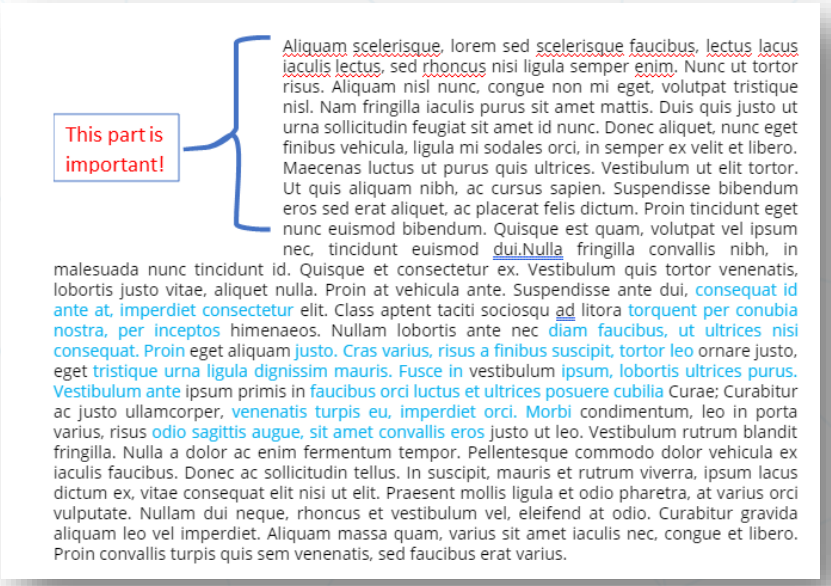


Technical Challenges – Floating Content



Aliquam scelerisque, lorem sed scelerisque faucibus, lectus lacus iaculis lectus, sed rhoncus nisi ligula semper enim. Nunc ut tortor risus. Aliquam nisl nunc, congue non mi eget, volutpat tristique nisl. Nam fringilla iaculis purus sit amet mattis. Duis quis justo ut urna sollicitudin feugiat sit amet id nunc. Donec aliquet, nunc eget finibus vehicula, ligula mi sodales orci, in semper ex velit et libero. Maecenas luctus ut purus quis ultrices. Vestibulum ut elit tortor. Ut quis aliquam nibh, ac cursus sapien. Suspendisse bibendum eros sed erat aliquet, ac placerat felis dictum. Proin tincidunt eget nunc euismod bibendum. Quisque est quam, volutpat vel ipsum nec, tincidunt euismod dui.Nulla fringilla convallis nibh, in malesuada nunc tincidunt id. Quisque et consectetur ex. Vestibulum quis tortor venenatis, lobortis justo vitae, aliquet nulla. Proin at vehicula ante. Suspendisse ante dui, elit. Class aptent taciti sociosqu ad litora himenaeos. Nullam lobortis ante nec eget aliquam ornare justo, eget vestibulum ipsum primis in Curae; Curabitur ac justo ullamcorper, condimentum, leo in porta varius, risus consequat id ante at, imperdiet consectetur torquent per conubia nostra, per inceptos diam faucibus, ut ultrices nisi consequat. Proin justo. Cras varius, risus a finibus suscipit, tortor leo tristique urna ligula dignissim mauris. Fusce in ipsum, lobortis ultrices purus. Vestibulum ante faucibus orci luctus et ultrices posuere cubilia venenatis turpis eu, imperdiet orci. Morbi odio sagittis augue, sit amet convallis eros justo ut leo. Vestibulum rutrum blandit fringilla. Nulla a dolor ac enim fermentum tempor. Pellentesque commodo dolor vehicula ex iaculis faucibus. Donec ac sollicitudin tellus. In suscipit, mauris et rutrum viverra, ipsum lacus dictum ex, vitae consequat elit nisi ut elit. Praesent mollis ligula et odio pharetra, at varius orci vulputate. Nullam dui neque, rhoncus et vestibulum vel, eleifend at odio. Curabitur gravida aliquam leo vel imperdiet. Aliquam massa quam, varius sit amet iaculis nec, congue et libero. Proin convallis turpis quis sem venenatis, sed faucibus erat varius.

MS Word 365 for Windows



Aliquam scelerisque, lorem sed scelerisque faucibus, lectus lacus iaculis lectus, sed rhoncus nisi ligula semper enim. Nunc ut tortor risus. Aliquam nisl nunc, congue non mi eget, volutpat tristique nisl. Nam fringilla iaculis purus sit amet mattis. Duis quis justo ut urna sollicitudin feugiat sit amet id nunc. Donec aliquet, nunc eget finibus vehicula, ligula mi sodales orci, in semper ex velit et libero. Maecenas luctus ut purus quis ultrices. Vestibulum ut elit tortor. Ut quis aliquam nibh, ac cursus sapien. Suspendisse bibendum eros sed erat aliquet, ac placerat felis dictum. Proin tincidunt eget nunc euismod bibendum. Quisque est quam, volutpat vel ipsum nec, tincidunt euismod dui.Nulla fringilla convallis nibh, in malesuada nunc tincidunt id. Quisque et consectetur ex. Vestibulum quis tortor venenatis, lobortis justo vitae, aliquet nulla. Proin at vehicula ante. Suspendisse ante dui, consequat id ante at, imperdiet consectetur elit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nullam lobortis ante nec diam faucibus, ut ultrices nisi consequat. Proin eget aliquam justo. Cras varius, risus a finibus suscipit, tortor leo ornare justo, eget tristique urna ligula dignissim mauris. Fusce in vestibulum ipsum, lobortis ultrices purus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Curabitur ac justo ullamcorper, venenatis turpis eu, imperdiet orci. Morbi condimentum, leo in porta varius, risus odio sagittis augue, sit amet convallis eros justo ut leo. Vestibulum rutrum blandit fringilla. Nulla a dolor ac enim fermentum tempor. Pellentesque commodo dolor vehicula ex iaculis faucibus. Donec ac sollicitudin tellus. In suscipit, mauris et rutrum viverra, ipsum lacus dictum ex, vitae consequat elit nisi ut elit. Praesent mollis ligula et odio pharetra, at varius orci vulputate. Nullam dui neque, rhoncus et vestibulum vel, eleifend at odio. Curabitur gravida aliquam leo vel imperdiet. Aliquam massa quam, varius sit amet iaculis nec, congue et libero. Proin convallis turpis quis sem venenatis, sed faucibus erat varius.

MS Word 365 Online

PDFTRON

OOXML Undefined Behavior

Small selection of required information missing from the OOXML specification:

- Float placement algorithm
- When do elements overlap?
- Page breaking algorithm
- Text category specification
- XLSX formula definitions
- Chart rendering information
- Small caps implementation
- Embedded html behaviour
- Color tinting and shading formulas
- Tab placement algorithm
- Inter-paragraph spacing algorithm
- Table cell border drawing algorithm
- Paragraph border drawing algorithm
- Overlarge item placement behaviour
- Line spacing algorithm

A Story About Animations

```
<p:timing>
    <p:tnLst>
        <p:par>
            <p:cTn id="1" dur="indefinite" restart="never" nodeType="tmRoot">
                <p:childTnLst>
                    <p:seq concurrent="1" nextAc="seek">
                        <p:cTn id="2" dur="indefinite" nodeType="mainSeq">
                            <p:childTnLst>
                                <p:par>
                                    <p:cTn id="3" fill="hold">
                                        <p:stCondLst>
                                            <p:cond delay="indefinite"/>
                                        </p:stCondLst>
                                        <p:childTnLst>
                                            <p:par>
                                                <p:cTn id="4" fill="hold">
                                                    <p:stCondLst>
                                                        <p:cond delay="0"/>
                                                    </p:stCondLst>
                                                    <p:childTnLst>
                                                        <p:par>
                                                            <p:cTn id="5" presetID="10" presetClass="entr"
                                                        presetSubtype="0" fill="hold" grpId="0" nodeType="clickEffect">
                                                                <p:stCondLst>
                                                                    <p:cond delay="0"/>
                                                                </p:stCondLst>
                                                                <p:childTnLst>
                                                                    <p:animEffect transition="in" filter="fade">
                                                                        <p:cBhvr>
                                                                            <p:cTn id="7" dur="500"/>
                                                                            <p:tgtEl>
                                                                                <p:spTgt spid="2"/>
                                                                            </p:tgtEl>
                                                                        </p:cBhvr>
                                                                    </p:animEffect>
                                                                </p:childTnLst>
```

A Story About Animations

According to ECMA 376:

fill: This attribute describes the fill type for the time node.

Not helpful.... But references `ST_TLTimeNodeFillType`.

This looks better!

This simple type specifies what modifications the effect leaves on the target element's properties when the effect ends.

Enumeration Value	Description
freeze (Freeze)	Freeze
hold (Hold)	Hold
remove (Remove)	Remove
transition (Transition)	Transition

...that's it?

Standards Don’t Guarantee Interoperability

- Office renderers will not converge towards “correct” behaviour
- Risk of different rendering will always be there

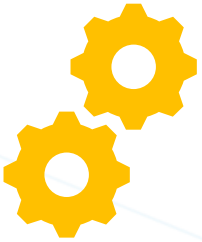
- Use PDF for reliable communication!

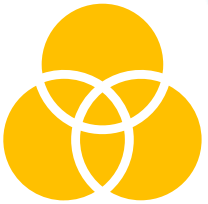
Roadmap

- Introduction
- Defining the problem
  - Viewing, sharing, archival, editing
- An in depth look
  - PDF for viewing, Office for editing
  - Office rendering is unreliable
  - Office rendering will never be reliable
  - PDF for sharing and archival!
- Next: Solutions
  - Combine the strengths of both formats
  -
  -
- Conclusion

Is Reliable Office Rendering Possible?

- No, but…
- PDF is good at the things that office is not
- We can create workflows that use the best of each format

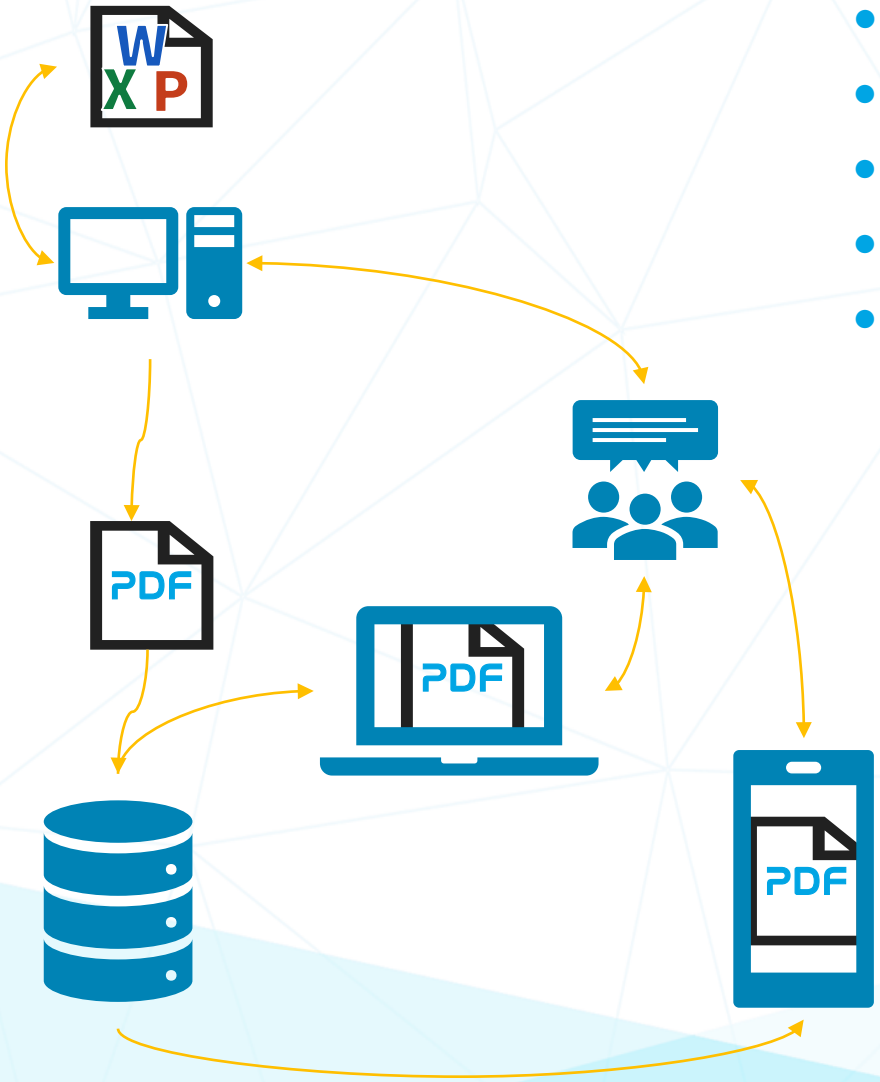




Roadmap

- Introduction
- Defining the problem
  - Viewing, sharing, archival, editing
- An in depth look
  - PDF for viewing, Office for editing
  - Office rendering is unreliable
  - Office rendering will never be reliable
  - PDF for sharing and archival!
- Next: Solutions
  - Combine the strengths of both formats
  - Workflow A: users convert to PDF
  -
- Conclusion

Workflow A: Users Convert to PDF



- Relies on policy rather than technology
- Conversion likely performed by MS Office
- Don’t know what version
- Don’t know what settings
- Inefficient

Roadmap

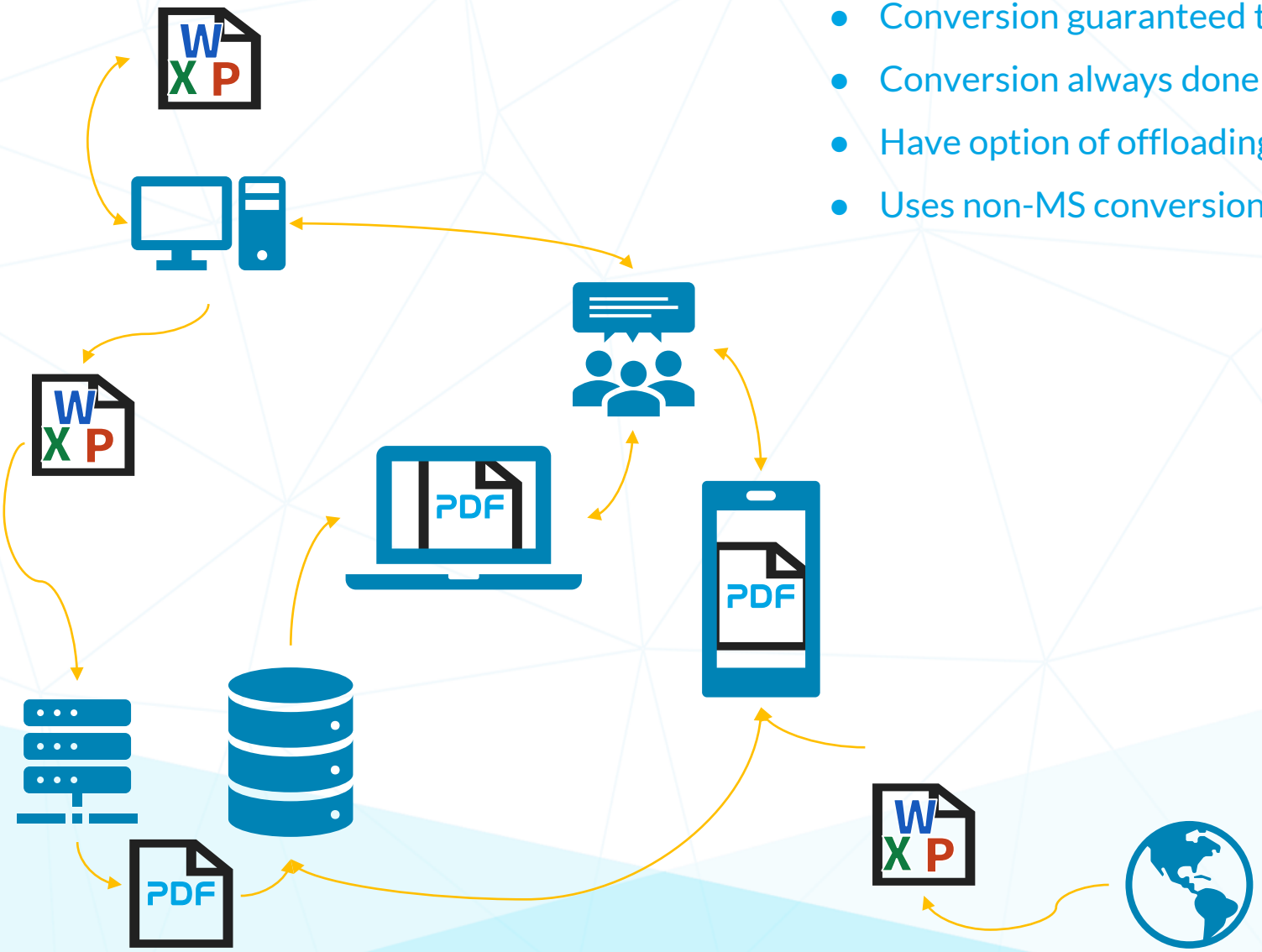
- Introduction
- Defining the problem
  - Viewing, sharing, archival, editing
- An in depth look
  - PDF for viewing, Office for editing
  - Office rendering is unreliable
  - Office rendering will never be reliable
  - PDF for sharing and archival!
- Next: Solutions
  - Combine the strengths of both formats
  - Workflow A: users convert to PDF
  - Workflow B: central conversion
- Conclusion

Why Not Use MS Office?

MS Office has a number of limitations:

- Cannot run on Linux servers
- Unreliable in Windows server environments
- Legally questionable in Windows server environments
- Can’t be embedded

Workflow B: Central Conversion



- Conversion guaranteed to happen
- Conversion always done the same way
- Have option of offloading work to clients
- Uses non-MS conversion software

Roadmap

- Introduction
- Defining the problem
  - Viewing, sharing, archival, editing
- An in depth look
  - PDF for viewing, Office for editing
  - Office rendering is unreliable
  - Office rendering will never be reliable
  - PDF for sharing and archival!
- Next: Solutions
  - Combine the strengths of both formats
  - Workflow A: users convert to PDF
  - Workflow B: central conversion service
- Conclusion

Roadmap

- Introduction
- Defining the problem
  - Viewing, sharing, archival, editing
- An in depth look
  - PDF for viewing, Office for editing
  - Office rendering is unreliable
  - Office rendering will never be reliable
  - PDF for sharing and archival!
- Next: Solutions
  - Combine the strengths of both formats
  - Workflow A: users convert to PDF
  - Workflow B: central conversion service
- Conclusion
  - OOXML/A ?

Thank You!

www.pdftron.com
info@pdftron.com