



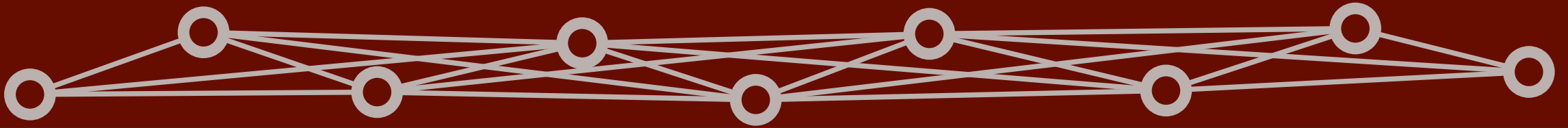
Backward  
compatibility  
as  
swimming against  
the tide

Bogusław Jackowski

**Bachotek, 3 V 2024**

# Once upon a time...

Once upon a time (more exactly, it was the end of the seventies) Marek & I were stinged by the idea of making fonts with the help of a computer. Marek, a student then, prepared a program for printing letters on a line printer.

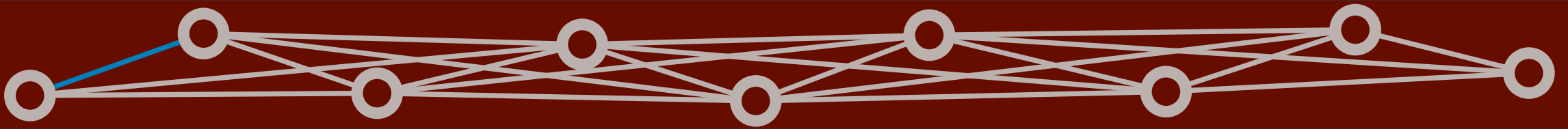




# Once upon a time...

Once upon a time (more exactly, it was the end of the seventies) Marek & I were stung by the idea of making fonts with the help of a computer. Marek, a student then, prepared a program for printing letters on a line printer.

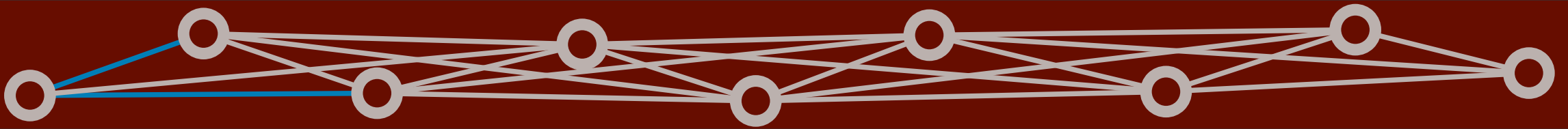
Marek prepared his program in 1976. A year later, prof. D.E. Knuth wrote a memo to himself describing the basic features of T<sub>E</sub>X.



# Once upon a time...

Once upon a time (more exactly, it was the end of the seventies) Marek & I were stung by the idea of making fonts with the help of a computer. Marek, a student then, prepared a program for printing letters on a line printer.

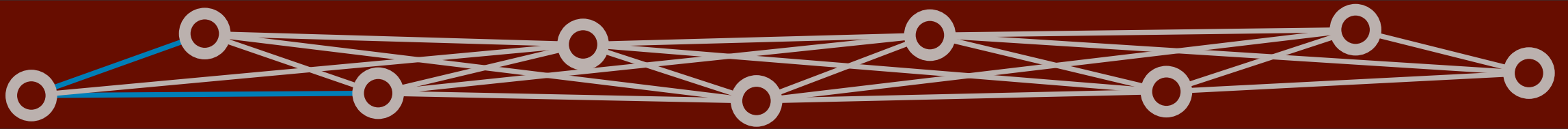
Marek prepared his program in 1976. A year later, prof. D.E. Knuth wrote a memo to himself describing the basic features of T<sub>E</sub>X. It is hard not to connect the dots...



# Once upon a time...

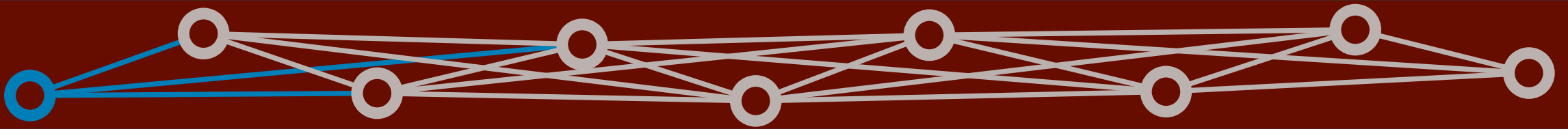
Once upon a time (more exactly, it was the end of the seventies) Marek & I were stung by the idea of making fonts with the help of a computer. Marek, a student then, prepared a program for printing letters on a line printer.

Marek prepared his program in 1976. A year later, prof. D.E. Knuth wrote a memo to himself describing the basic features of T<sub>E</sub>X. It is hard not to connect the dots...



# What about backward compatibility in the T<sub>E</sub>X font realm?

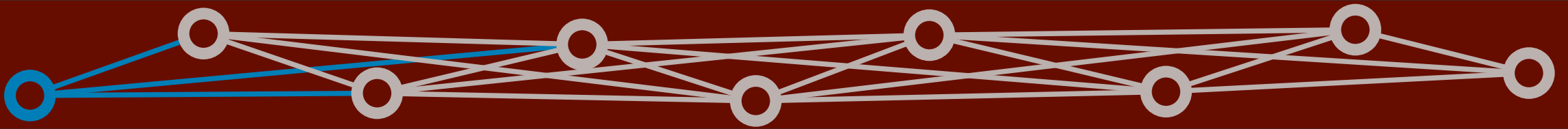
More seriously, although T<sub>E</sub>X and METAFONT were published already in 1977, their more stable versions appeared in 1982 (T<sub>E</sub>X) and 1984 (METAFONT); both programs received 8-bit functionality in 1990 and then they were announced to be “frozen”.





# What about backward compatibility in the T<sub>E</sub>X font realm?

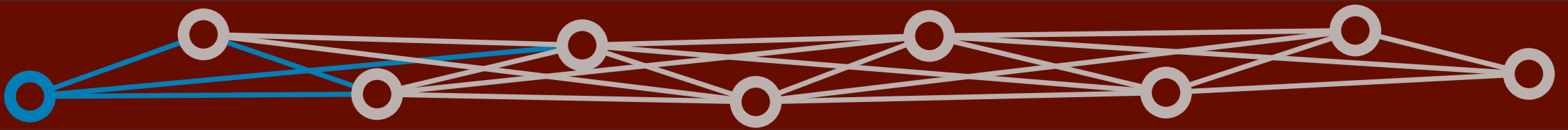
More seriously, although T<sub>E</sub>X and METAFONT were published already in 1977, their more stable versions appeared in 1982 (T<sub>E</sub>X) and 1984 (METAFONT); both programs received 8-bit functionality in 1990 and then they were announced to be “frozen”. At the end of the *cm85.bug* file prof. D.E. Knuth writes:



# What about backward compatibility in the T<sub>E</sub>X font realm?

More seriously, although T<sub>E</sub>X and METAFONT were published already in 1977, their more stable versions appeared in 1982 (T<sub>E</sub>X) and 1984 (METAFONT); both programs received 8-bit functionality in 1990 and then they were announced to be “frozen”. At the end of the *cm85.bug* file prof. D.E. Knuth writes:

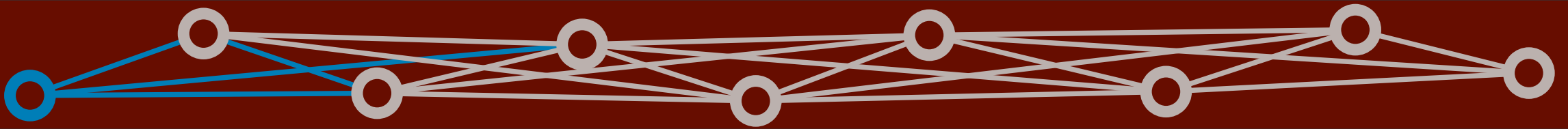
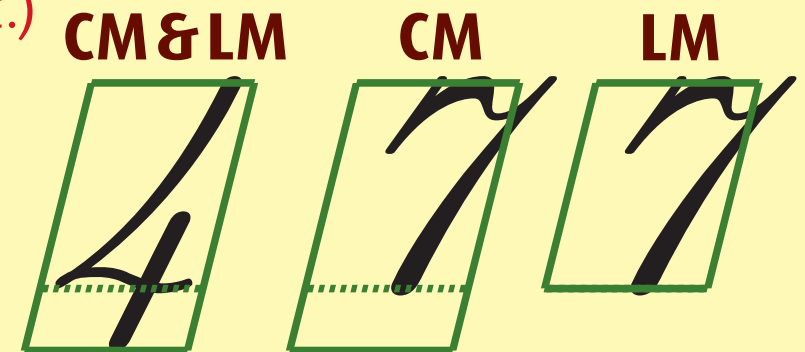
-----Here I draw the line with respect to further changes  
(I sincerely believe there won't be any more!)  
I absolutely guarantee that the TFM files will never change again.  
(Otherwise I would consider zeroing the depth of italic 7,  
which I admit is strange... we can live with it.)



# What about backward compatibility in the T<sub>E</sub>X font realm?

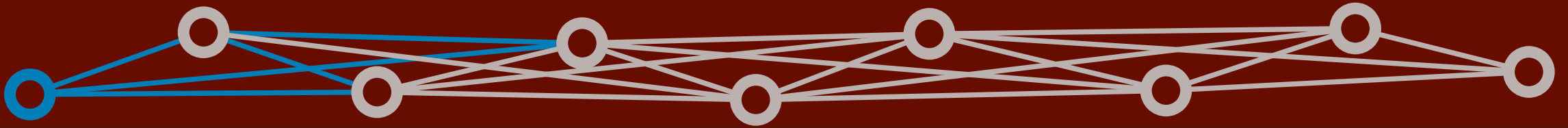
More seriously, although T<sub>E</sub>X and METAFONT were published already in 1977, their more stable versions appeared in 1982 (T<sub>E</sub>X) and 1984 (METAFONT); both programs received 8-bit functionality in 1990 and then they were announced to be “frozen”. At the end of the *cm85.bug* file prof. D.E. Knuth writes:

-----Here I draw the line with respect to further changes  
(I sincerely believe there won't be any more!)  
I absolutely guarantee that the TFM files will never change again.  
(Otherwise I would consider zeroing the depth of italic 7,  
which I admit is strange... we can live with it.)

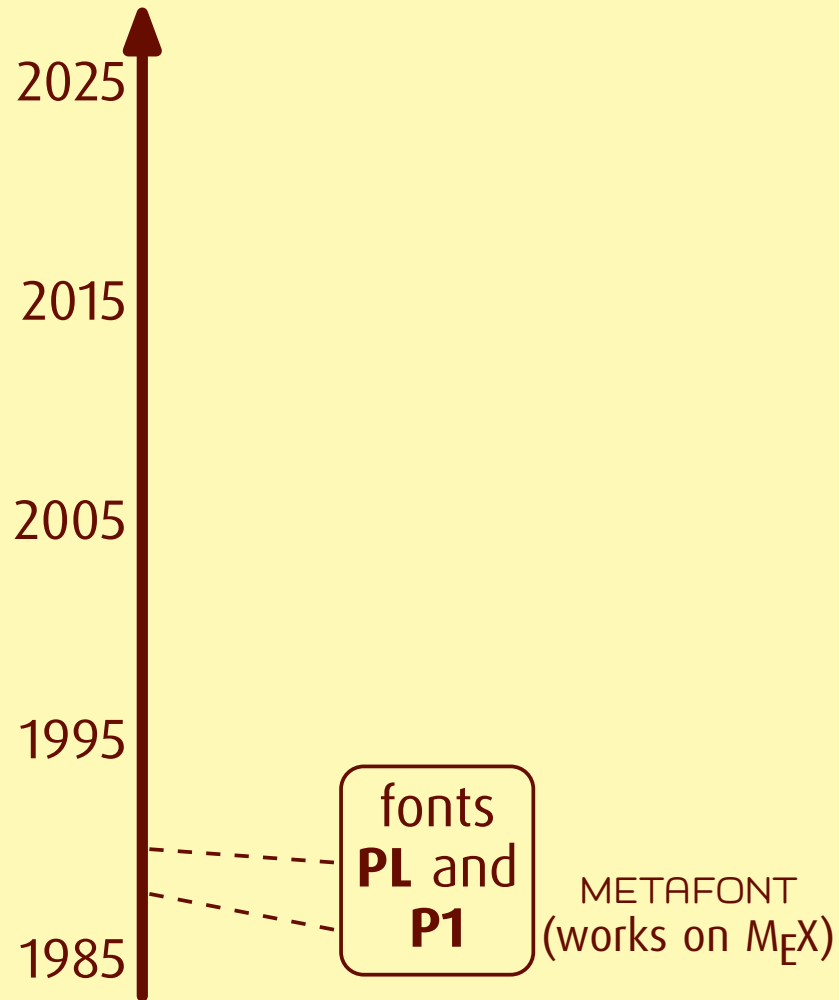


# What about backward compatibility in GUST e-Foundry products

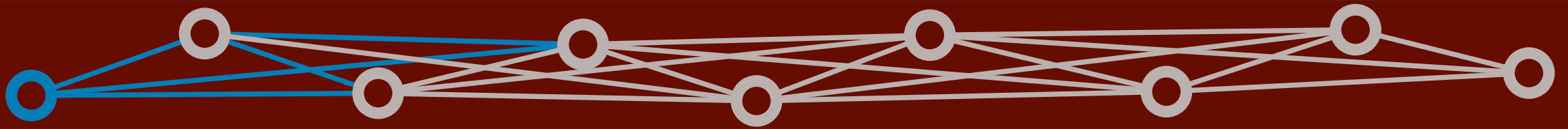
Let's start with a bit of history – not to boast about our achievements, but to highlight the changes in font technology that occurred as we proceeded with subsequent projects.



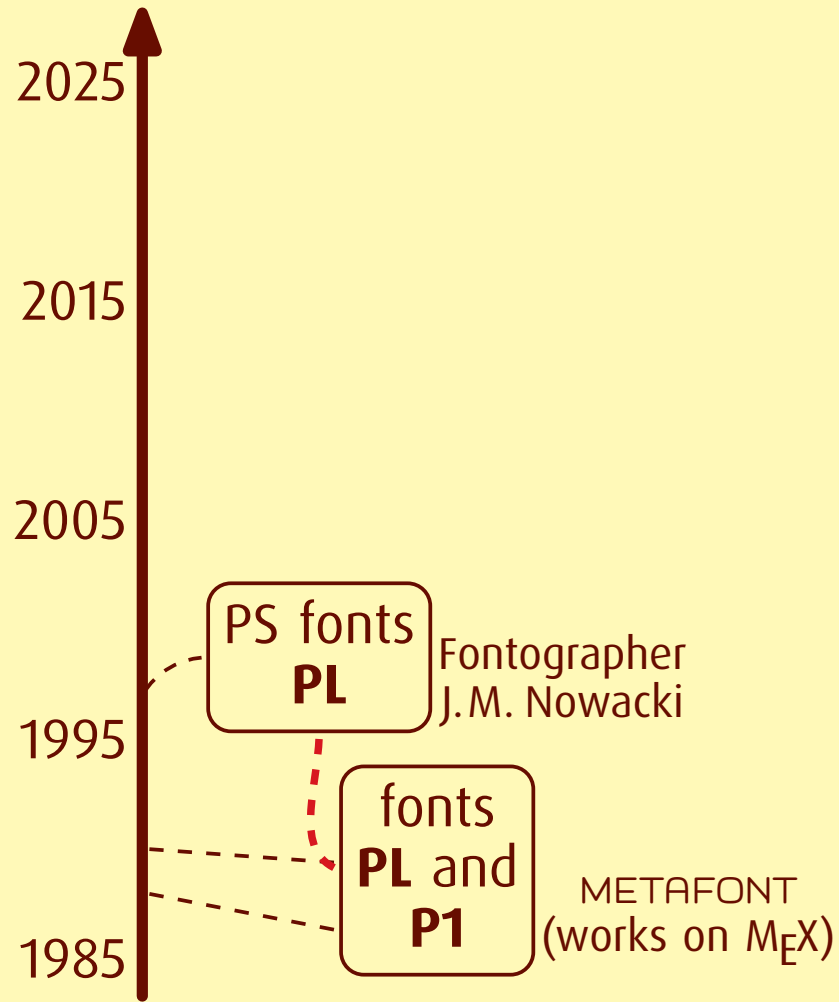
# What about backward compatibility in GUST e-Foundry products



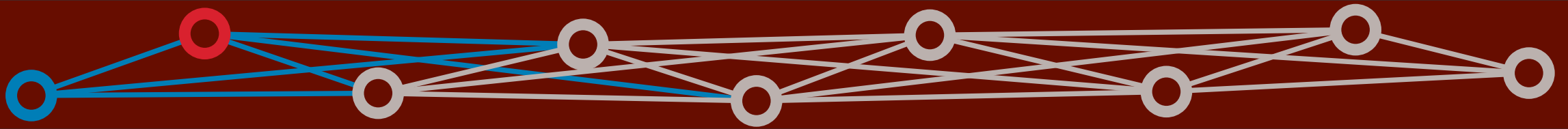
Let's start with a bit of history – not to boast about our achievements, but to highlight the changes in font technology that occurred as we proceeded with subsequent projects.



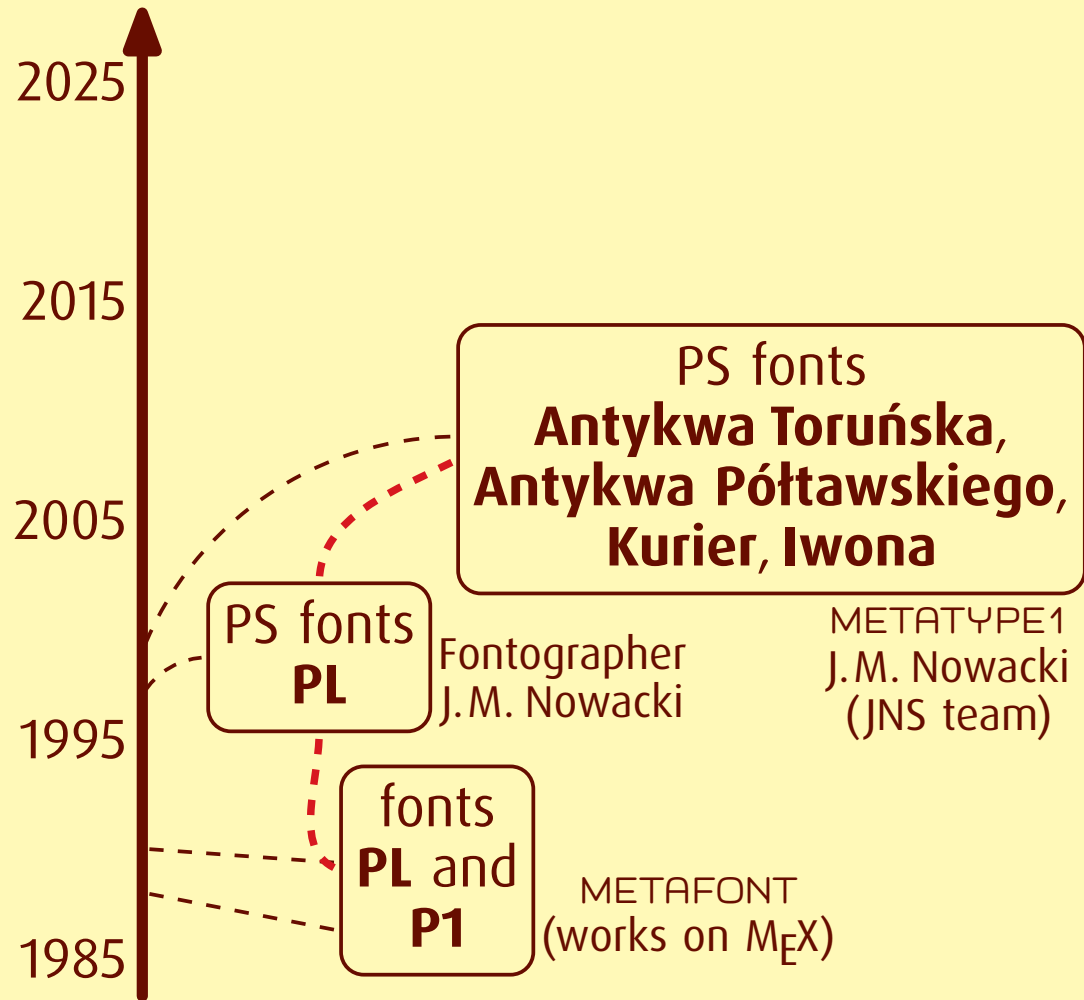
# What about backward compatibility in GUST e-Foundry products



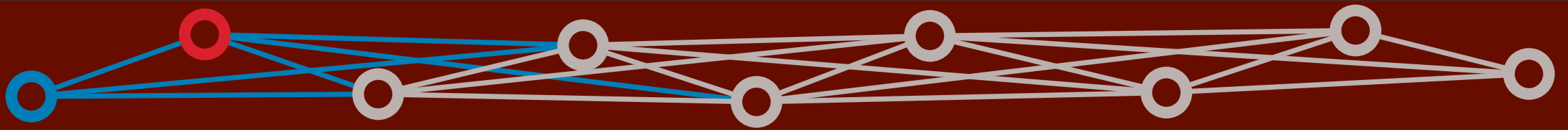
Let's start with a bit of history – not to boast about our achievements, but to highlight the changes in font technology that occurred as we proceeded with subsequent projects.



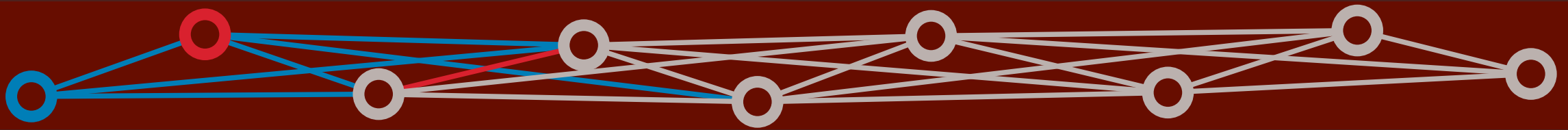
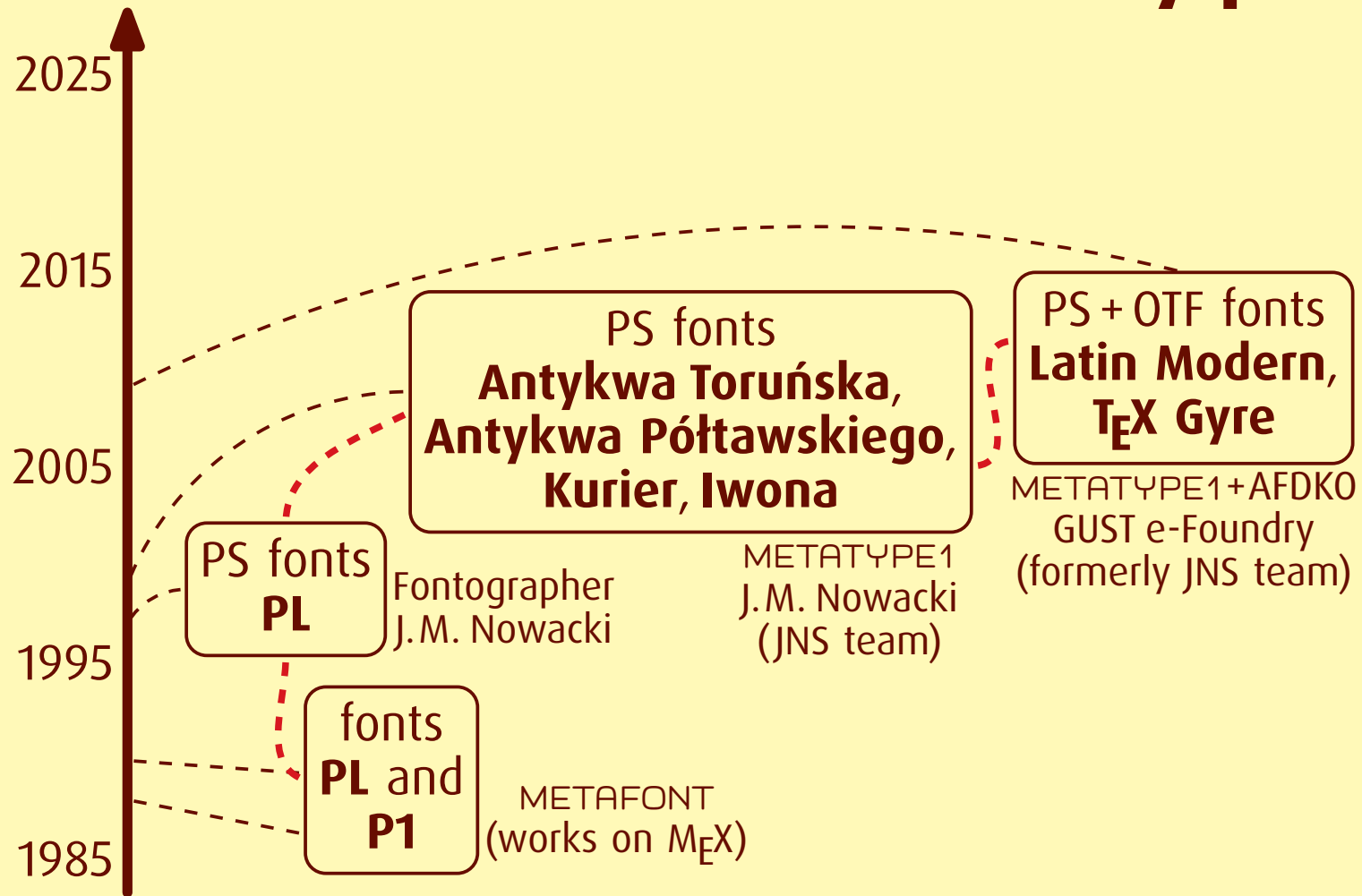
# What about backward compatibility in GUST e-Foundry products



Let's start with a bit of history – not to boast about our achievements, but to highlight the changes in font technology that occurred as we proceeded with subsequent projects.

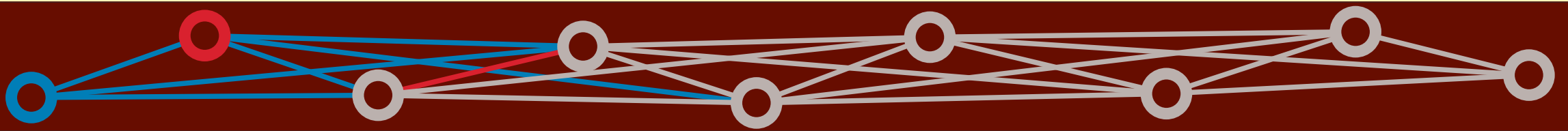
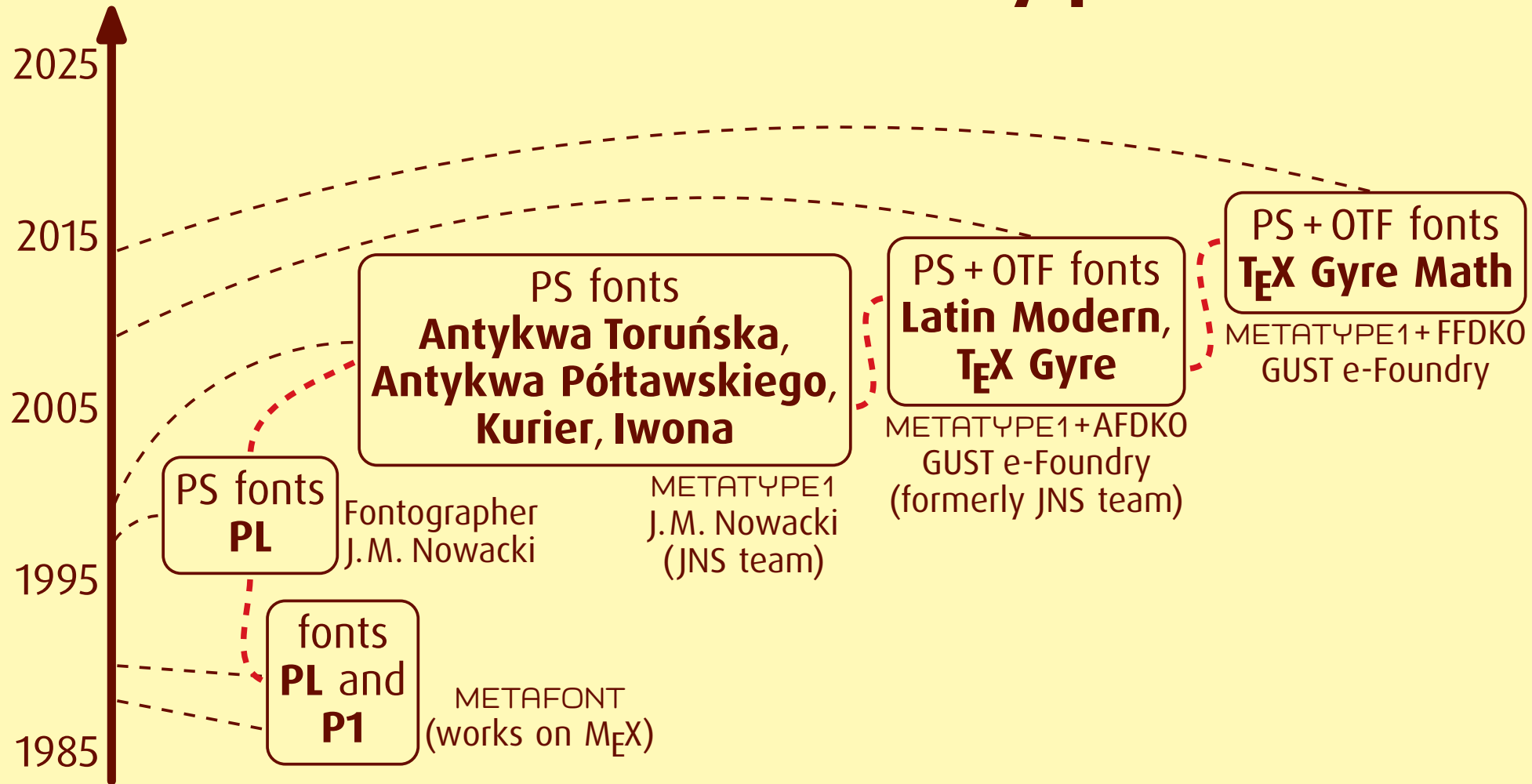


# What about backward compatibility in GUST e-Foundry products

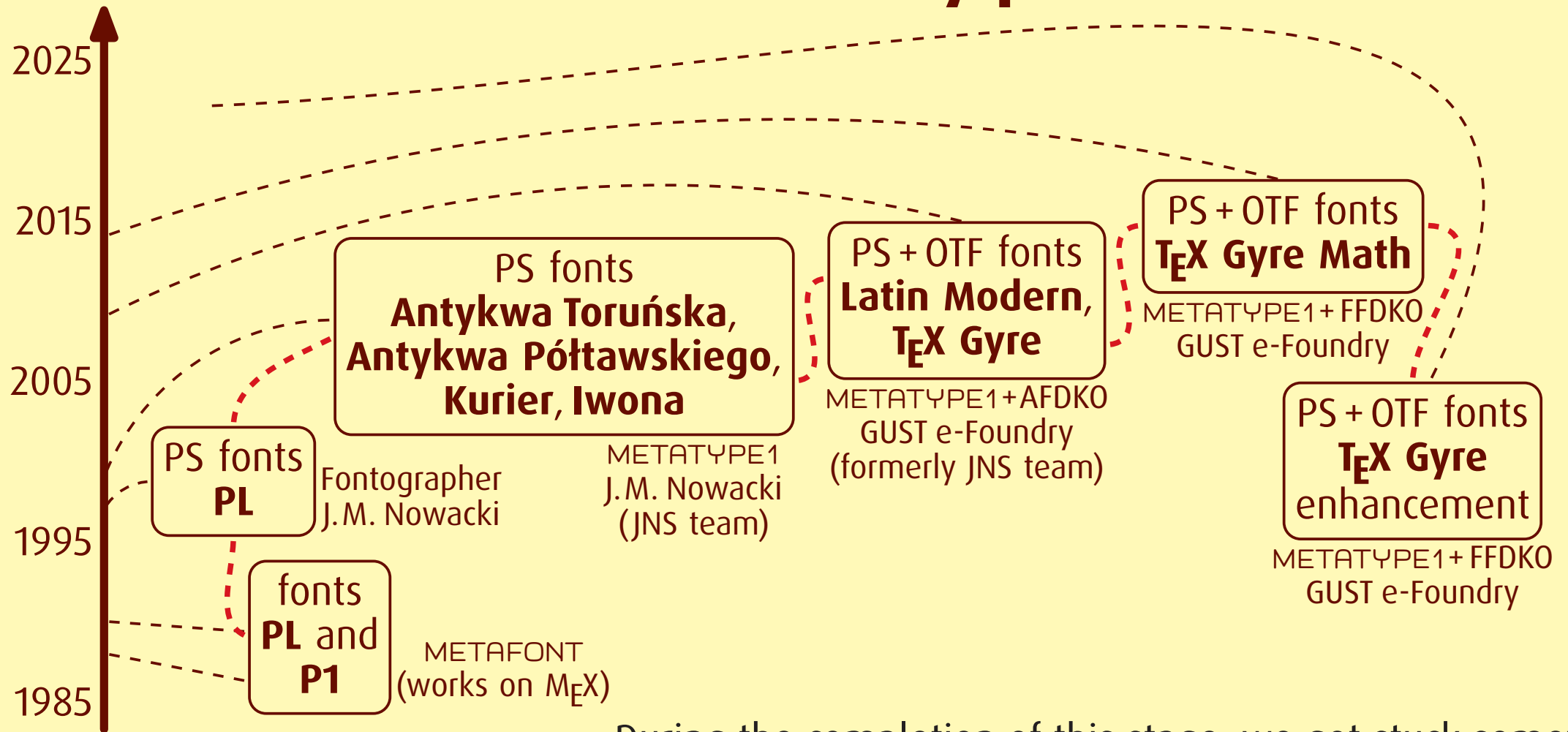




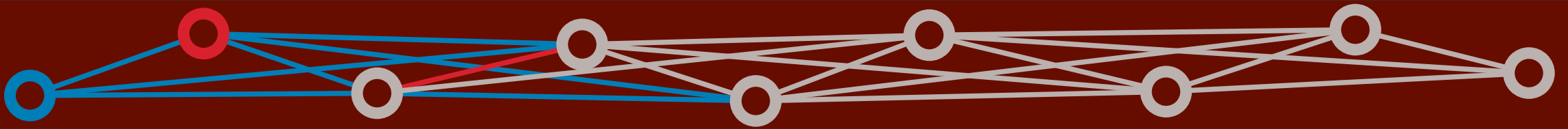
# What about backward compatibility in GUST e-Foundry products



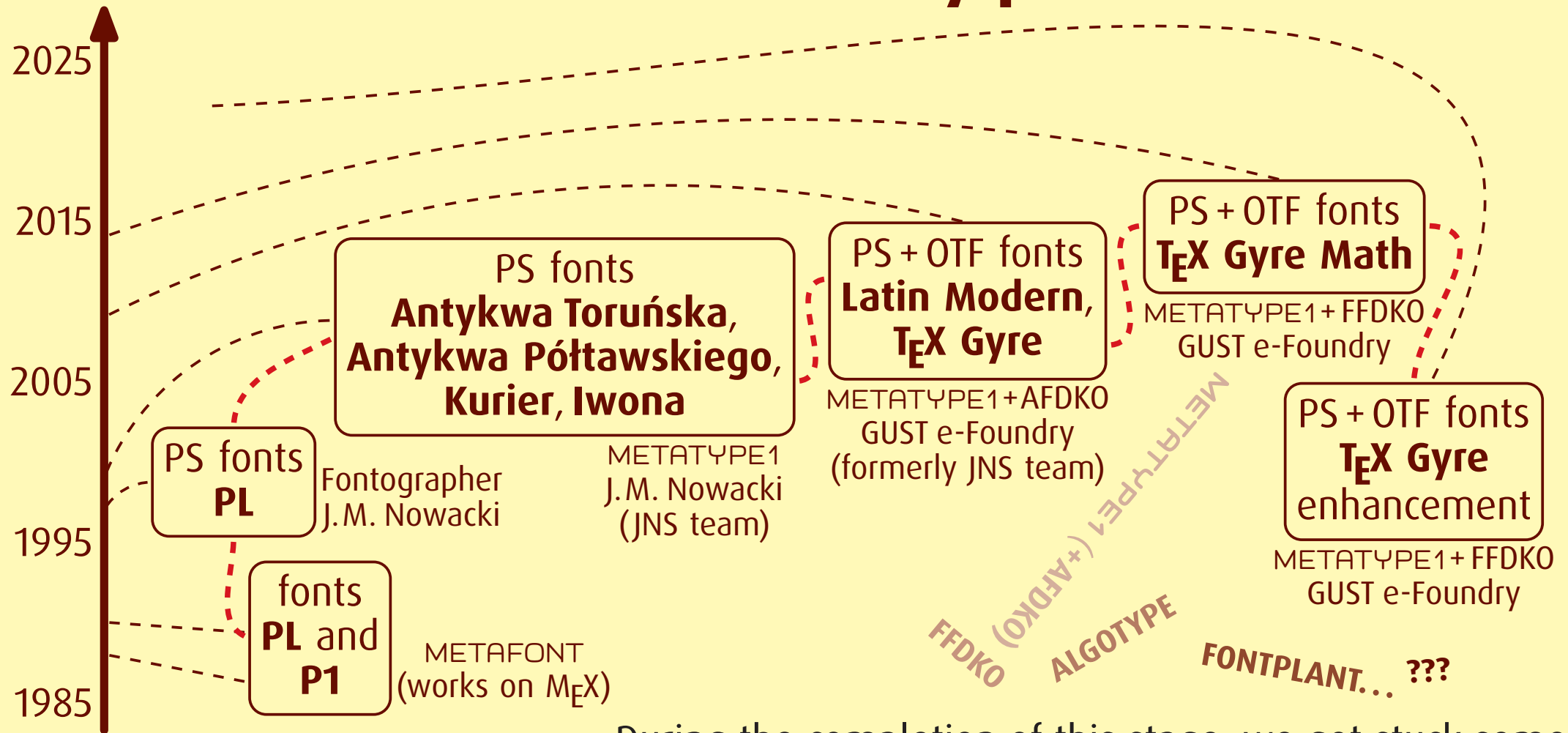
# What about backward compatibility in GUST e-Foundry products



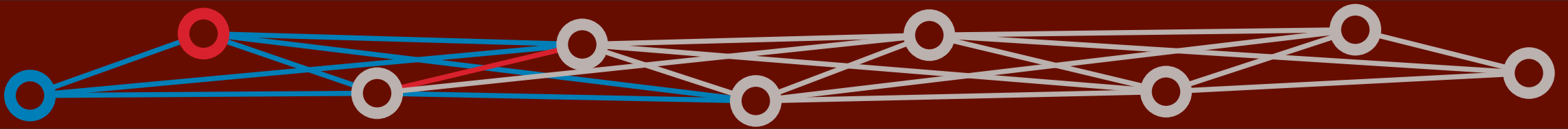
During the completion of this stage, we got stuck somewhat.



# What about backward compatibility in GUST e-Foundry products



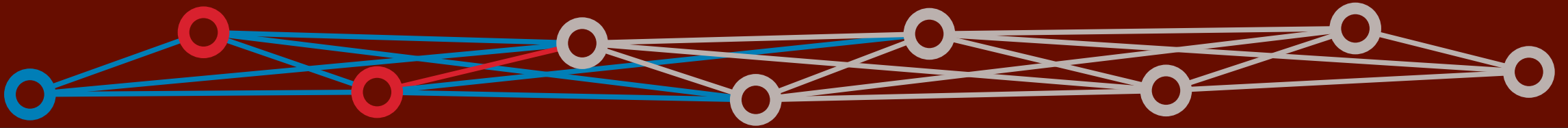
During the completion of this stage, we got stuck somewhat.





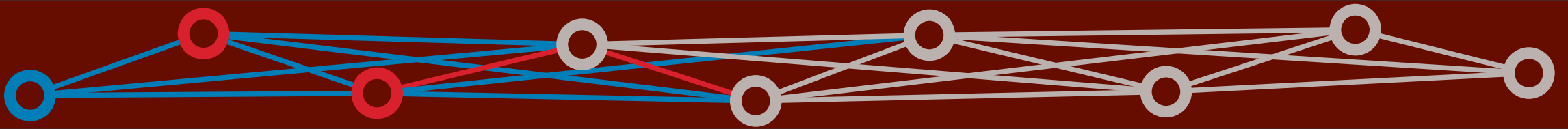
# Backward compatibility: who cares?

- Our presumption: statistically, nobody.



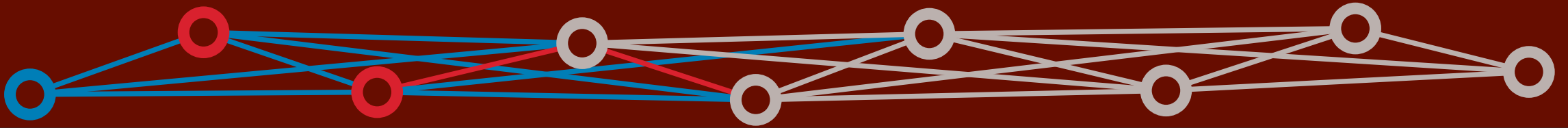
# Backward compatibility: who cares?

- Our presumption: statistically, nobody.
- Exceptionally, prof. D.E. Knuth.



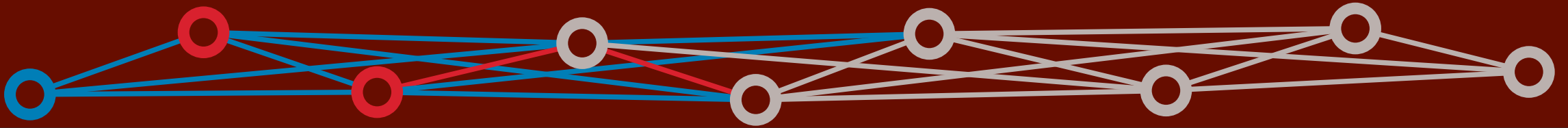
# Backward compatibility: who cares?

- Our presumption: statistically, nobody.
- Exceptionally, prof. D.E. Knuth.
- We (GUST e-Foundry) are trying to care...



# Backward compatibility: who cares?

- Our presumption: statistically, nobody.
- Exceptionally, prof. D.E. Knuth.
- We (GUST e-Foundry) are trying to care...
- It seems that large corporations do care about incompatibility instead.





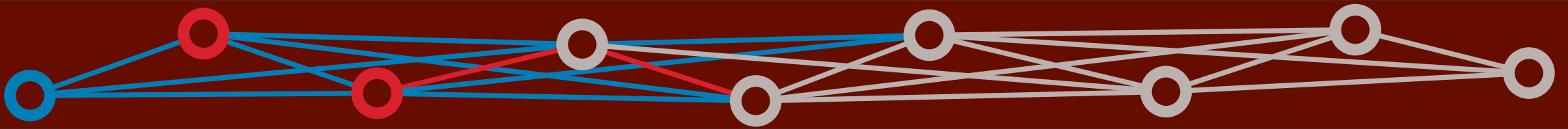
# Backward compatibility: who cares?

- Our presumption: statistically, nobody.
- Exceptionally, prof. D.E. Knuth.
- We (GUST e-Foundry) are trying to care...
- It seems that large corporations do care about incompatibility instead.

Thanks a lot for letting me use  
this photo in my presentation.

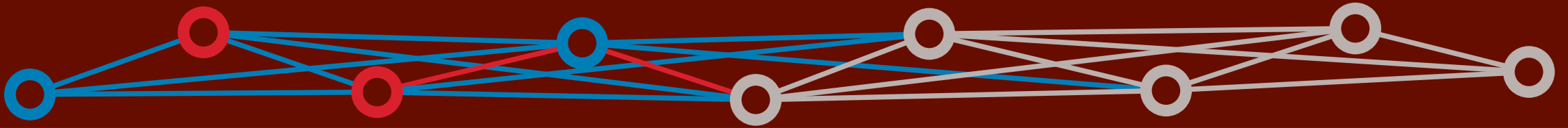


*A B<sub>ach</sub>oT<sub>E</sub>X participant captured at the moment she noticed the system updating again.*



# Backward compatibility: who cares?

- Our presumption: statistically, nobody.
- Exceptionally, prof. D.E. Knuth.
- We (GUST e-Foundry) are trying to care...
- It seems that large corporations do care about incompatibility instead.  
Just two examples:

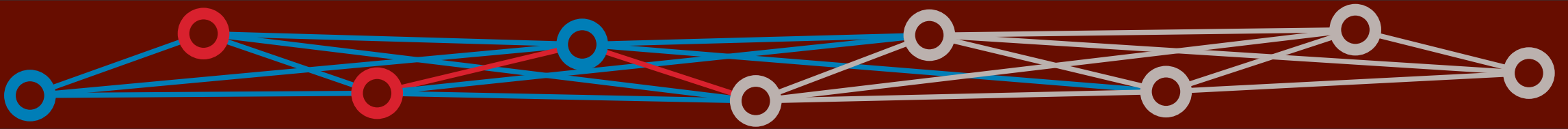


# Backward compatibility: who cares?

- Our presumption: statistically, nobody.
- Exceptionally, prof. D.E. Knuth.
- We (GUST e-Foundry) are trying to care...
- It seems that large corporations do care about incompatibility instead.  
Just two examples:

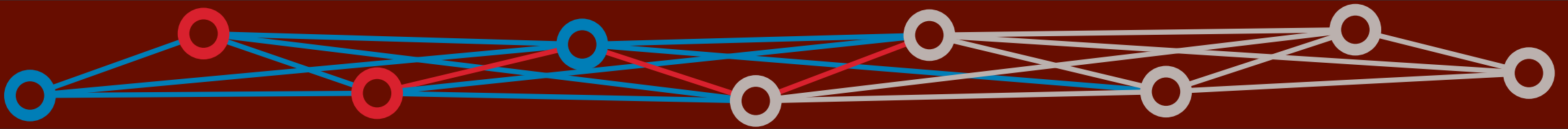
It's not accurate to say that programs designed for Windows 3 cannot run on Windows 10. Personally, I still use on Windows 10 a program compiled in 1989. While it is possible, it does require some effort, which the creators of the operating system intentionally chose not to undertake.

Similarly, I use CorelDRAW 8 (1997) under Windows 10, although it requires additional software (VMware). Once again, the software vendors chose not to provide such a facility for old customers.

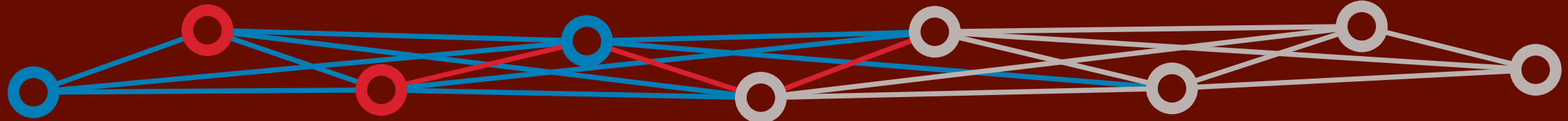


# Backward compatibility: who cares?

- Our presumption: statistically, nobody.
- Exceptionally, prof. D.E. Knuth.
- We (GUST e-Foundry) are trying to care...
- It seems that large corporations do care about incompatibility instead.
- Therefore, we aimed to achieve a level of backward compatibility that, while not perfect, is acceptable. (I'm afraid that the number given by Rysio was too accurate.)

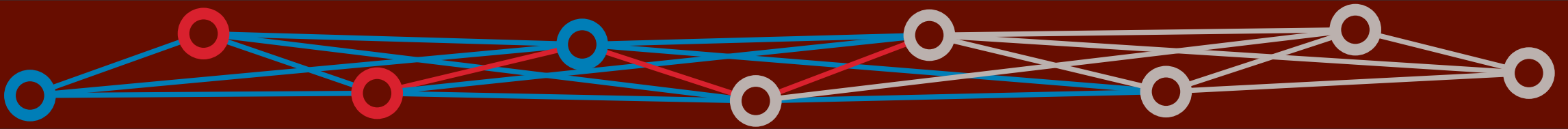


# Compatibility issues we had to cope with: a few examples



# Compatibility issues we had to cope with: a few examples

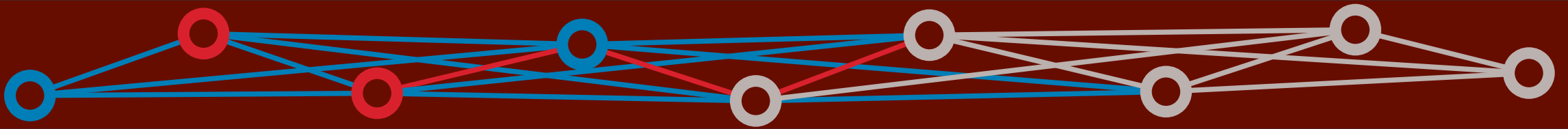
- To cut a long story short, the primary and most significant difference between the “pioneer times” and now is the increased complexity of the OTF format compared to the  $\text{T}_{\text{E}}\text{X}$  bitmap/TFM and PostScript Type1 formats, as well as the lack of reliable supporting software.



# Compatibility issues we had to cope with: a few examples

- To cut a long story short, the primary and most significant difference between the “pioneer times” and now is the increased complexity of the OTF format compared to the  $\text{T}_{\text{E}}\text{X}$  bitmap/TFM and PostScript Type1 formats, as well as the lack of reliable supporting software.
- We were happy to learn that a *Python* script for *FontForge* is available, *sfddiff.pe*<sup>\*</sup>, that compares two font files which may be in any format that *FontForge* can read.

<sup>\*</sup> <https://raw.githubusercontent.com/fontforge/fontforge/master/fontforgeexe/sfddiff.pe>



# Compatibility issues we had to cope with: a few examples

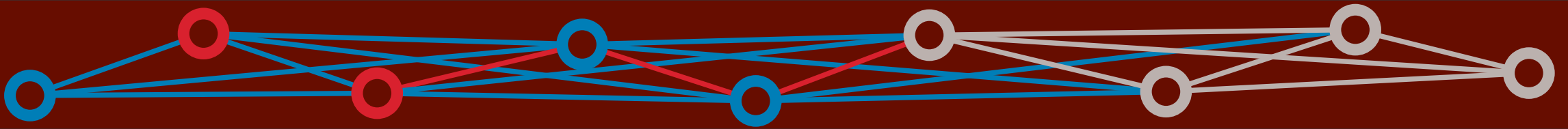
- To cut a long story short, the primary and most significant difference between the “pioneer times” and now is the increased complexity of the OTF format compared to the T<sub>E</sub>X bitmap/TFM and PostScript Type1 formats, as well as the lack of reliable supporting software.
- We were happy to learn that a *Python* script for *FontForge* is available, *sfddiff.pe*<sup>\*</sup>, that compares two font files which may be in any format that *FontForge* can read. The results of comparison, however, not always are satisfactory:

**No kerning between 'dcroat.sc' and atilde.sc in  
C:/BachoTeX/test-texgyreadventor-bold.otf  
whilst it is -30 in C:/BachoTeX/qagb.pfb**

...

**No kerning between 'dcroat.sc' and atilde.sc in  
C:/BachoTeX/qagb.pfb  
whilst it is -30 in C:/BachoTeX/test-texgyreadventor-bold.otf**

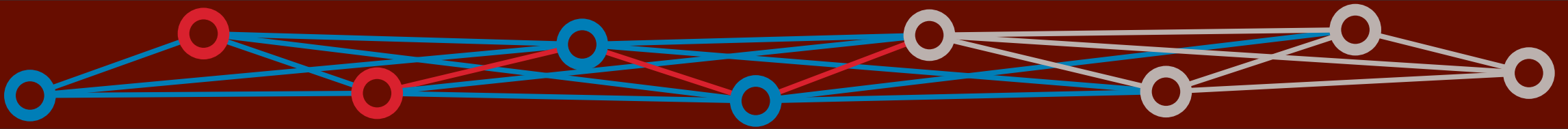
<sup>\*</sup> <https://raw.githubusercontent.com/fontforge/fontforge/master/fontforgeexe/sfddiff.pe>





# Compatibility issues we had to cope with: a few examples

- To cut a long story short, the primary and most significant difference between the “pioneer times” and now is the increased complexity of the OTF format compared to the  $\TeX$  bitmap/TFM and PostScript Type1 formats, as well as the lack of reliable supporting software.
- In certain tests, we can rely on the tried-and-true software (*tftopl*, *pltotf*, *dvitype*, etc.) – for example, for checking TFM files. However, even in these cases, surprises may still arise.



# Compatibility issues we had to cope with: a few examples

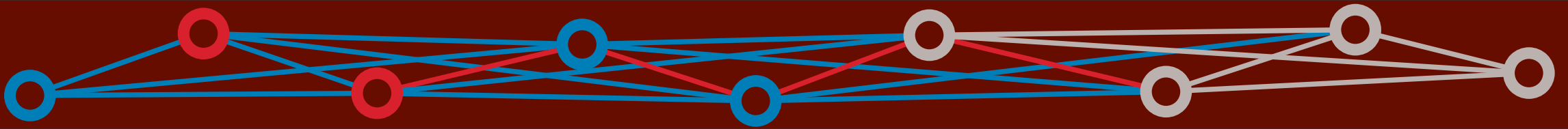
- To cut a long story short, the primary and most significant difference between the “pioneer times” and now is the increased complexity of the OTF format compared to the T<sub>E</sub>X bitmap/TFM and PostScript Type1 formats, as well as the lack of reliable supporting software.
- In certain tests, we can rely on the tried-and-true software (*tftopl*, *pltotf*, *dvitype*, etc.) – for example, for checking TFM files. However, even in these cases, surprises may still arise.

```
(CHARACTER C 6  
  (CHARWD R 0.660001)  
  (CHARHT R 0.673999)  
  )  
  ...  
(CHARACTER C 8  
  (CHARWD R 0.660001)  
  (CHARHT R 0.673999)  
  )
```

**Left column:** excerpt from the output of *tftopl* for the *qx-qbkb-sc.tfm* file (i.e., TG Bonum bold small caps) generated by METAPOST in 2009.

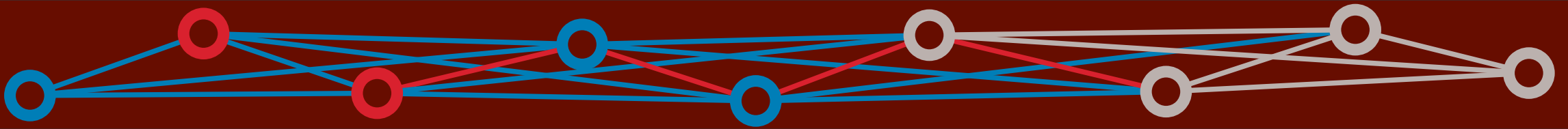
**Right column:** the corresponding excerpt for a TFM newly generated from 2009 sources.

**Only two bytes differ – this can't happen...**



# Compatibility issues we had to cope with: a few examples

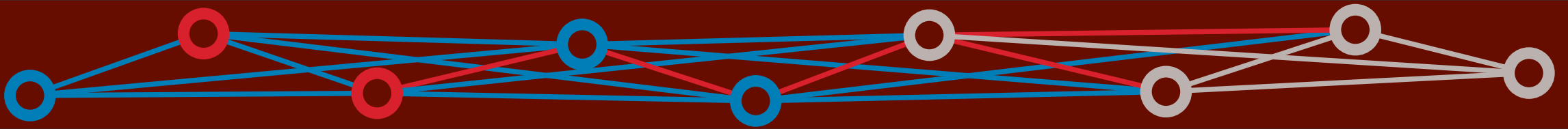
- To cut a long story short, the primary and most significant difference between the “pioneer times” and now is the increased complexity of the OTF format compared to the  $\TeX$  bitmap/TFM and PostScript Type1 formats.
- Fortunately, there exist reliable tools that can be used in the process of testing and comparing font. For us, Lua $\TeX$  was a great help – very many thanks Hans, Taco, & Hartmut!



# Compatibility issues we had to cope with: a few examples

- To cut a long story short, the primary and most significant difference between the “pioneer times” and now is the increased complexity of the OTF format compared to the T<sub>E</sub>X bitmap/TFM and PostScript Type1 formats.
- Fortunately, there exist reliable tools that can be used in the process of testing and comparing font. For us, LuaT<sub>E</sub>X was a great help – very many thanks Hans, Taco, & Hartmut!

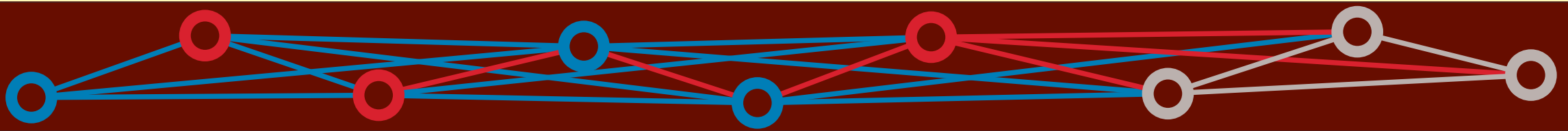
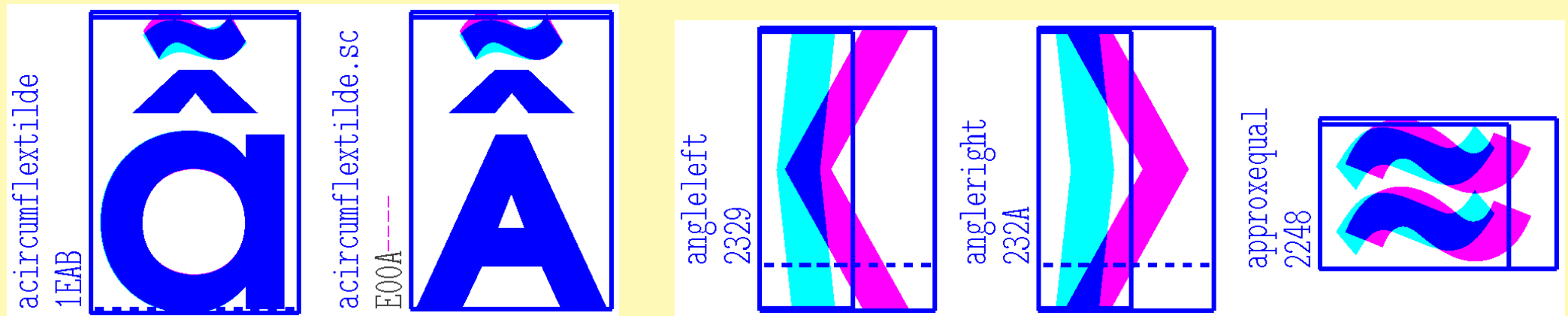
```
\nopagenumbers
\def\caron{\char"030C}\def\cedilla{\char"0327}
\def\tcedilla{\char"0163}\def\Tcedilla{\char"0162}
\def\@#1 {\font\A="[#1]:mode=node;+ccmp;+mark;+mkmk" at 32pt \A
  t\cedilla\caron\ t\caron\cedilla\ \tcedilla\caron \hskip1mm
  T\cedilla\caron\ T\caron\cedilla\ \Tcedilla\caron \hskip1mm
  \vrule width1pt \hskip1mm }
\@texgyreschola-bold.otf \@texgyreschola-bolditalic.otf
\@texgyreschola-italic.otf \@texgyreschola-regular.otf
\end
```





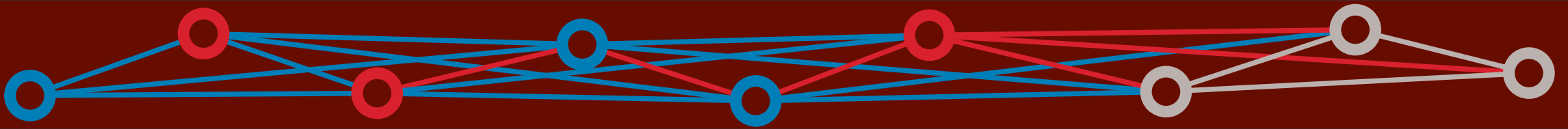
# Compatibility issues we had to cope with: a few examples

- To cut a long story short, the primary and most significant difference between the “pioneer times” and now is the increased complexity of the OTF format compared to the  $\TeX$  bitmap/TFM and PostScript Type1 formats.
- Fortunately, there exist reliable tools that can be used in the process of testing and comparing font. For us, Lua $\TeX$  was a great help – very many thanks Hans, Taco, & Hartmut!



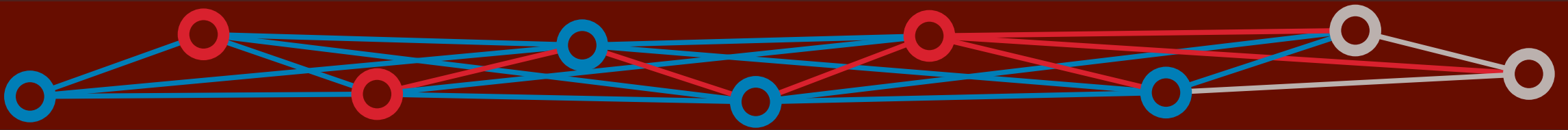
# Compatibility issues we had to cope with: a few examples

- To cut a long story short, the primary and most significant difference between the “pioneer times” and now is the increased complexity of the OTF format compared to the  $\text{T}_{\text{E}}\text{X}$  bitmap/TFM and PostScript Type1 formats.
- Anyway, testing and comparing fonts seem to garner little interest nowadays, resulting in a limited selection of relevant tools.



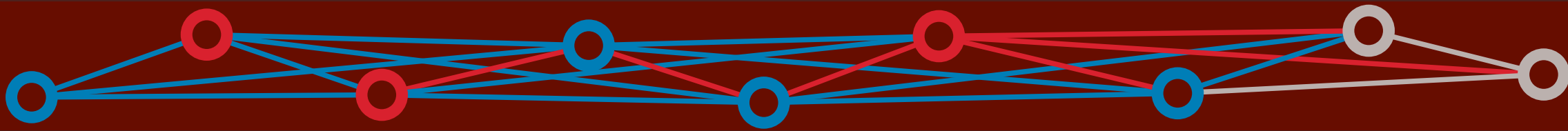
# Compatibility issues we had to cope with: a few examples

- To cut a long story short, the primary and most significant difference between the “pioneer times” and now is the increased complexity of the OTF format compared to the  $\text{T}_{\text{E}}\text{X}$  bitmap/TFM and PostScript Type1 formats.
- Anyway, testing and comparing fonts seem to garner little interest nowadays, resulting in a limited selection of relevant tools. So, why are we swimming against the tide? Why are we chasing the unattainable? It’s all because of the primary objective of the current GUST e-Foundry project: to publish the sources of the GUST e-Foundry fonts released thus far, along with the sources of the font-making engine and the relevant **documentation**.



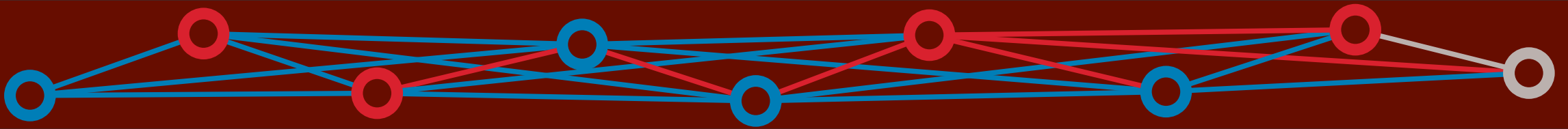


# Our agenda



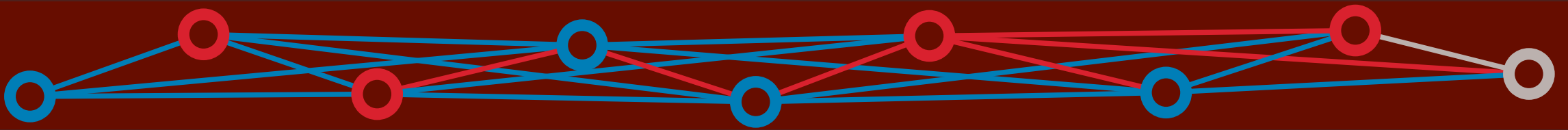
# Our agenda

- In the near future, we'd like to add the sources of Fontplant to the GUST GIT repository, along with the sources of the four T<sub>E</sub>X Gyre fonts we recently enhanced: *Adventor*, *Pagella*, *Schola*, and *Termes*.



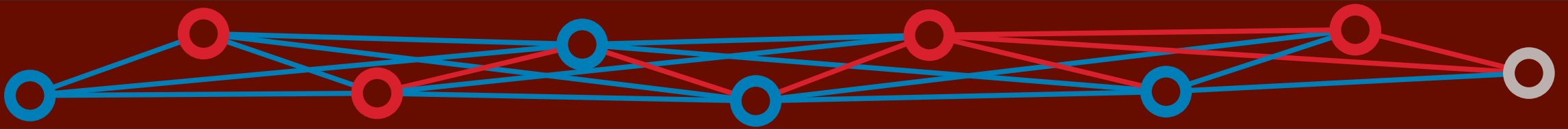
# Our agenda

- In the near future, we'd like to add the sources of Fontplant to the GUST GIT repository, along with the sources of the four T<sub>E</sub>X Gyre fonts we recently enhanced: *Adventor*, *Pagella*, *Schola*, and *Termes*.
- The next step would involve adding the sources of the remaining T<sub>E</sub>X Gyre fonts (*Bonum*, *Cursor*, *Heros*, *Chorus*) to the repository, possibly with slight adjustments to accommodate the new engine. No further enhancements are anticipated.



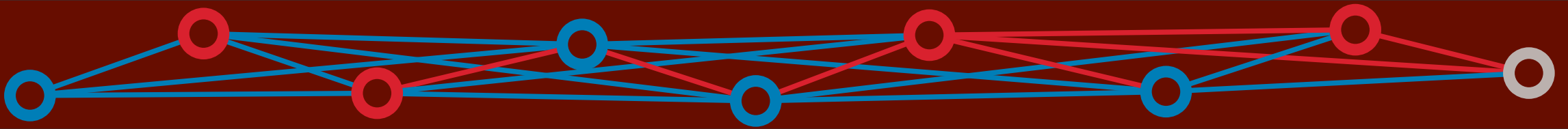
# Our agenda

- In the near future, we'd like to add the sources of Fontplant to the GUST GIT repository, along with the sources of the four T<sub>E</sub>X Gyre fonts we recently enhanced: *Adventor*, *Pagella*, *Schola*, and *Termes*.
- The next step would involve adding the sources of the remaining T<sub>E</sub>X Gyre fonts (*Bonum*, *Cursor*, *Heros*, *Chorus*) to the repository, possibly with slight adjustments to accommodate the new engine. No further enhancements are anticipated.
- Providing the sources for *Latin Modern* would be the obligatory next step. While we consider adding anchors, we are somewhat reluctant at the moment. We prioritize providing sources over improving the collection.



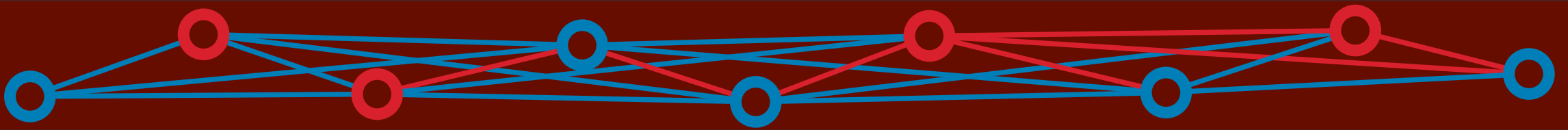
# Our agenda

- In the near future, we'd like to add the sources of Fontplant to the GUST GIT repository, along with the sources of the four T<sub>E</sub>X Gyre fonts we recently enhanced: *Adventor*, *Pagella*, *Schola*, and *Termes*.
- The next step would involve adding the sources of the remaining T<sub>E</sub>X Gyre fonts (*Bonum*, *Cursor*, *Heros*, *Chorus*) to the repository, possibly with slight adjustments to accommodate the new engine. No further enhancements are anticipated.
- Providing the sources for *Latin Modern* would be the obligatory next step. While we consider adding anchors, we are somewhat reluctant at the moment. We prioritize providing sources over improving the collection.
- We're undecided on what to do with the sources of T<sub>E</sub>X Gyre math fonts, but it may be feasible with the assistance of Piotr Strzelczyk, who played the first fiddle in the GUST e-Foundry ensemble for this project.



# Our agenda

- In the near future, we'd like to add the sources of Fontplant to the GUST GIT repository, along with the sources of the four T<sub>E</sub>X Gyre fonts we recently enhanced: *Adventor*, *Pagella*, *Schola*, and *Termes*.
- The next step would involve adding the sources of the remaining T<sub>E</sub>X Gyre fonts (*Bonum*, *Cursor*, *Heros*, *Chorus*) to the repository, possibly with slight adjustments to accommodate the new engine. No further enhancements are anticipated.
- Providing the sources for *Latin Modern* would be the obligatory next step. While we consider adding anchors, we are somewhat reluctant at the moment. We prioritize providing sources over improving the collection.
- We're undecided on what to do with the sources of T<sub>E</sub>X Gyre math fonts, but it may be feasible with the assistance of Piotr Strzelczyk, who played the first fiddle in the GUST e-Foundry ensemble for this project.
- Finally, we're considering publishing our sources of *Antykwa Półtawskiego*. While it may seem unnecessary given the recent release of the excellent font collection *Półtawski Nowy* by Ania Wieluńska, Borys Kosmyńka, and Mateusz Machalski, publishing sources for fully parameterized programmable outline fonts seems reasonable.



# BACHOTEX 2025

**PLEASE COME – AND BRING OTHERS**