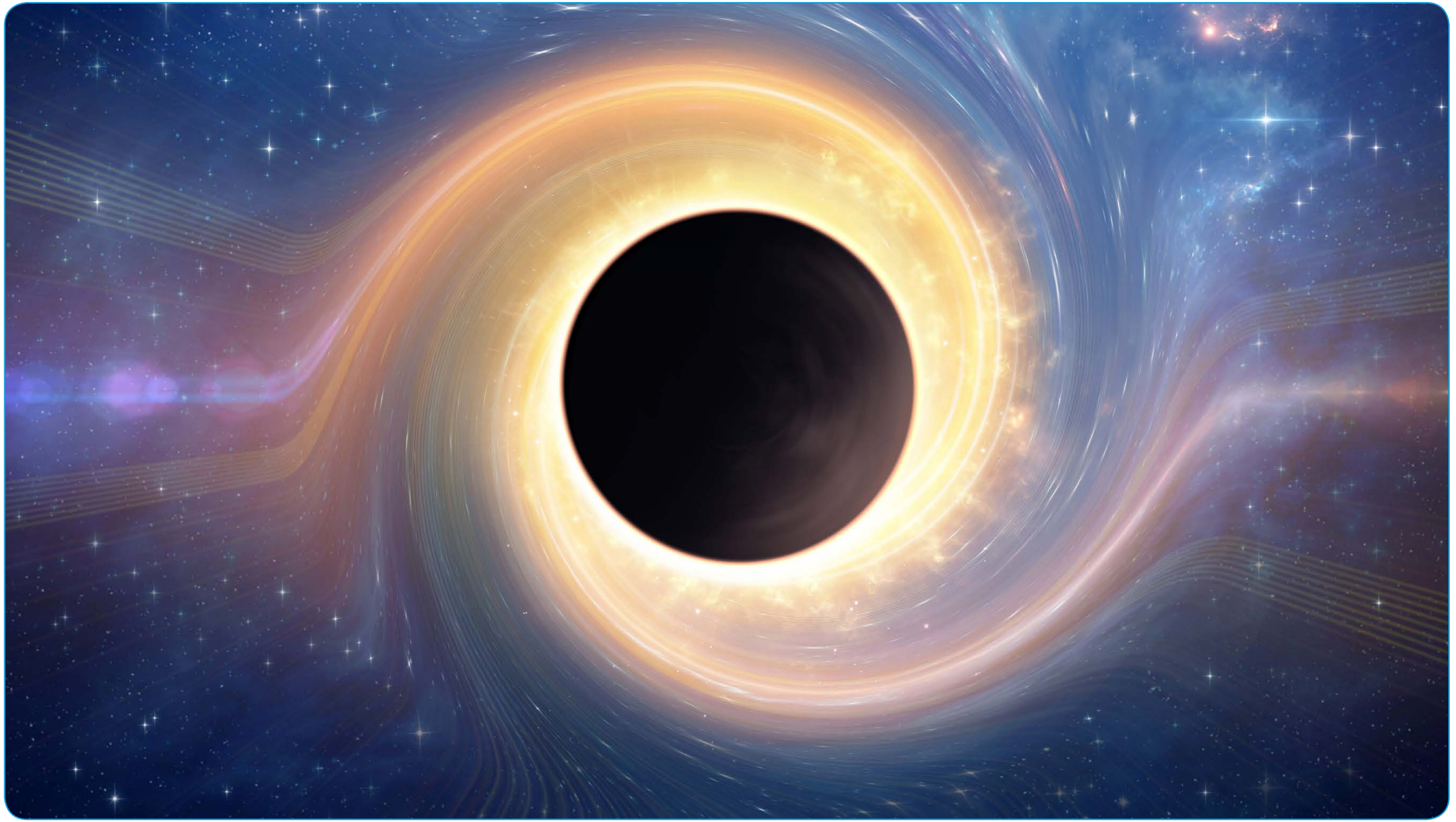
GHENT UNIVERSITY

Dr. David Fernández Duque

Ghent University

21 OCTOBER 2020



The Black Hole Theorem

LEAP SECONDS

LEAP SECONDS

Up to now, we have added 27 leap seconds.

UTC VS. UNIX

	11:59PM	11:60PM	12:00AM

UTC VS. UNIX

	11:59PM	11:60PM	12:00AM
Unix			

UTC VS. UNIX

	11:59PM	11:60PM	12:00AM
Unix	1.000.000		

UTC VS. UNIX

	11:59PM	11:60PM	12:00AM
Unix	1.000.000	1.000.000	

UTC VS. UNIX

	11:59PM	11:60PM	12:00AM
Unix	1.000.000	1.000.000	1.000.001

UTC VS. UNIX

	11:59PM	11:60PM	12:00AM
Unix	1.000.000	1.000.000	1.000.001
UTC			

UTC VS. UNIX

	11:59PM	11:60PM	12:00AM
Unix	1.000.000	1.000.000	1.000.001
UTC	1.000.000		

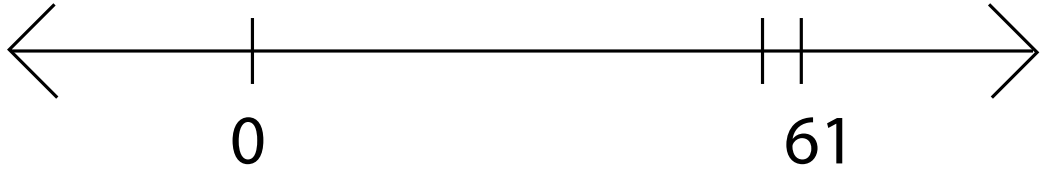
UTC VS. UNIX

	11:59PM	11:60PM	12:00AM
Unix	1.000.000	1.000.000	1.000.001
UTC	1.000.000	1.000.001	

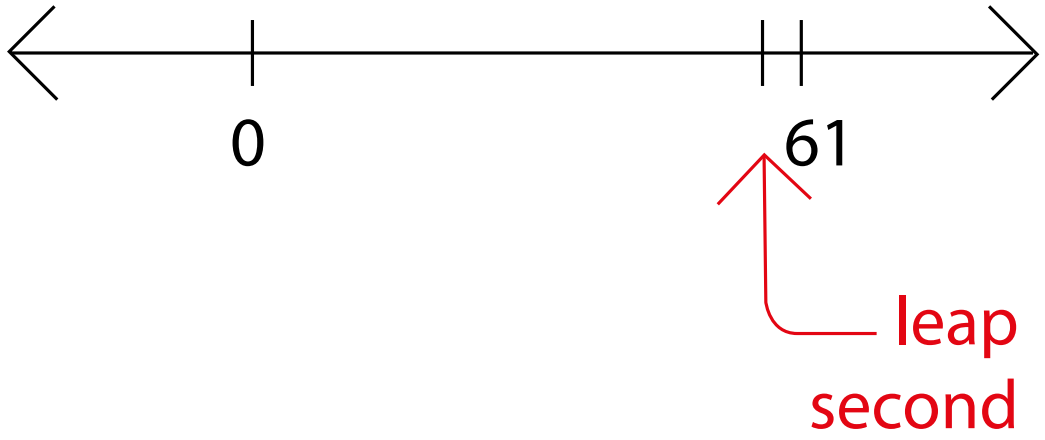
UTC VS. UNIX

	11:59PM	11:60PM	12:00AM
Unix	1.000.000	1.000.000	1.000.001
UTC	1.000.000	1.000.001	1.000.002

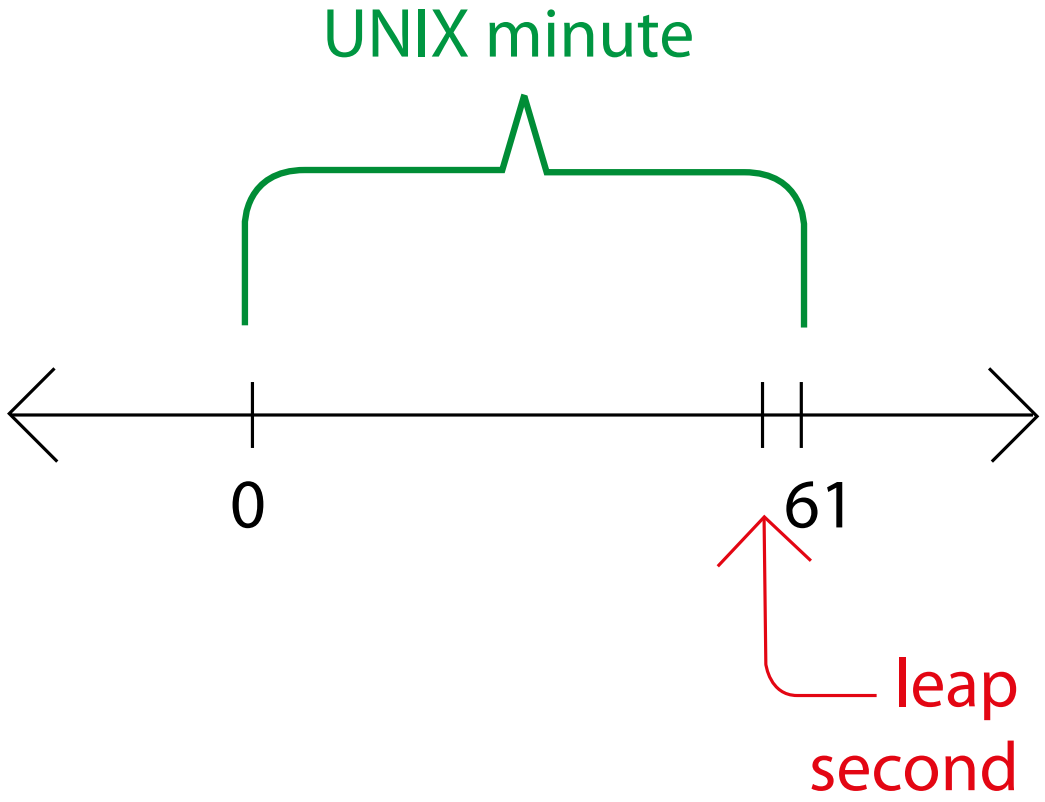
FORMALTIME: A STANDARIZED REPRESENTATION OF DURATIONS



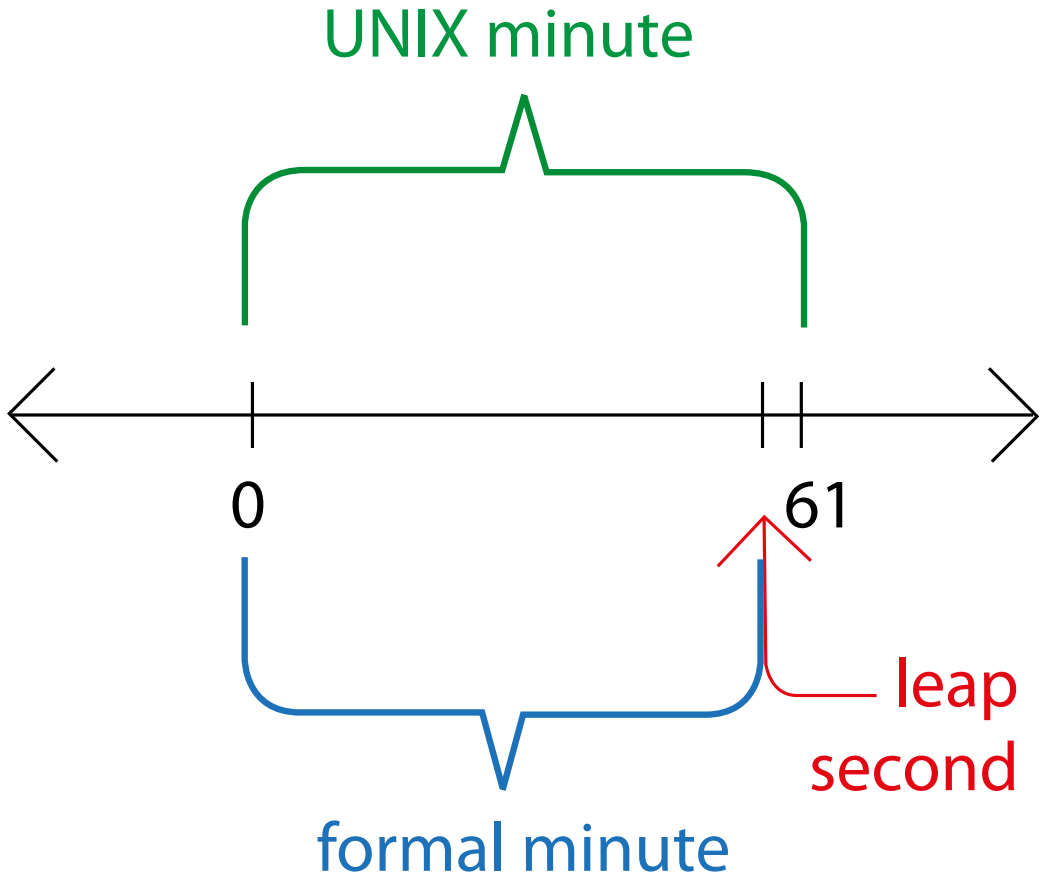
FORMALTIME: A STANDARIZED REPRESENTATION OF DURATIONS



FORMALTIME: A STANDARIZED REPRESENTATION OF DURATIONS

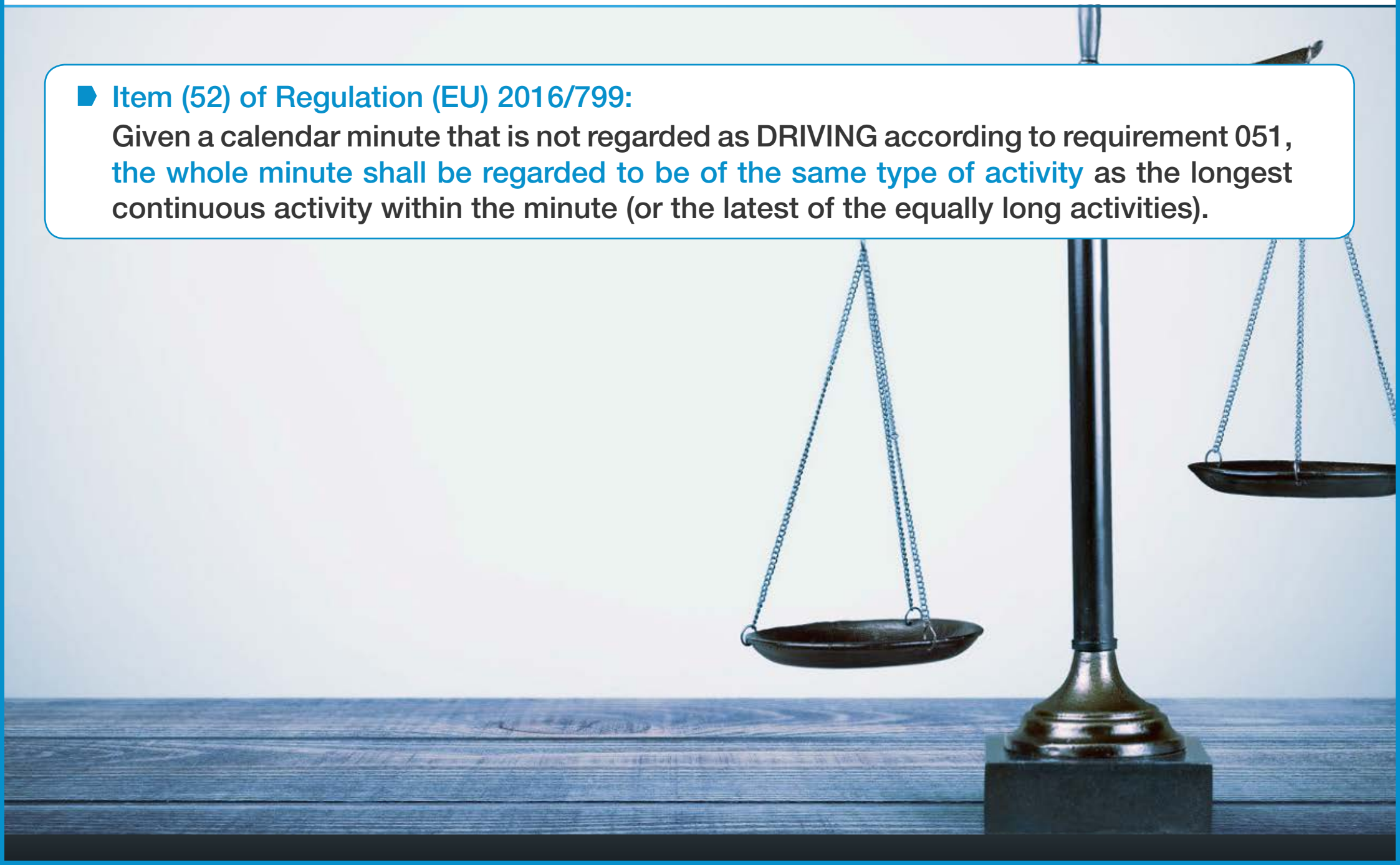


FORMALTIME: A STANDARIZED REPRESENTATION OF DURATIONS



UTC AND TACHOGRAPS

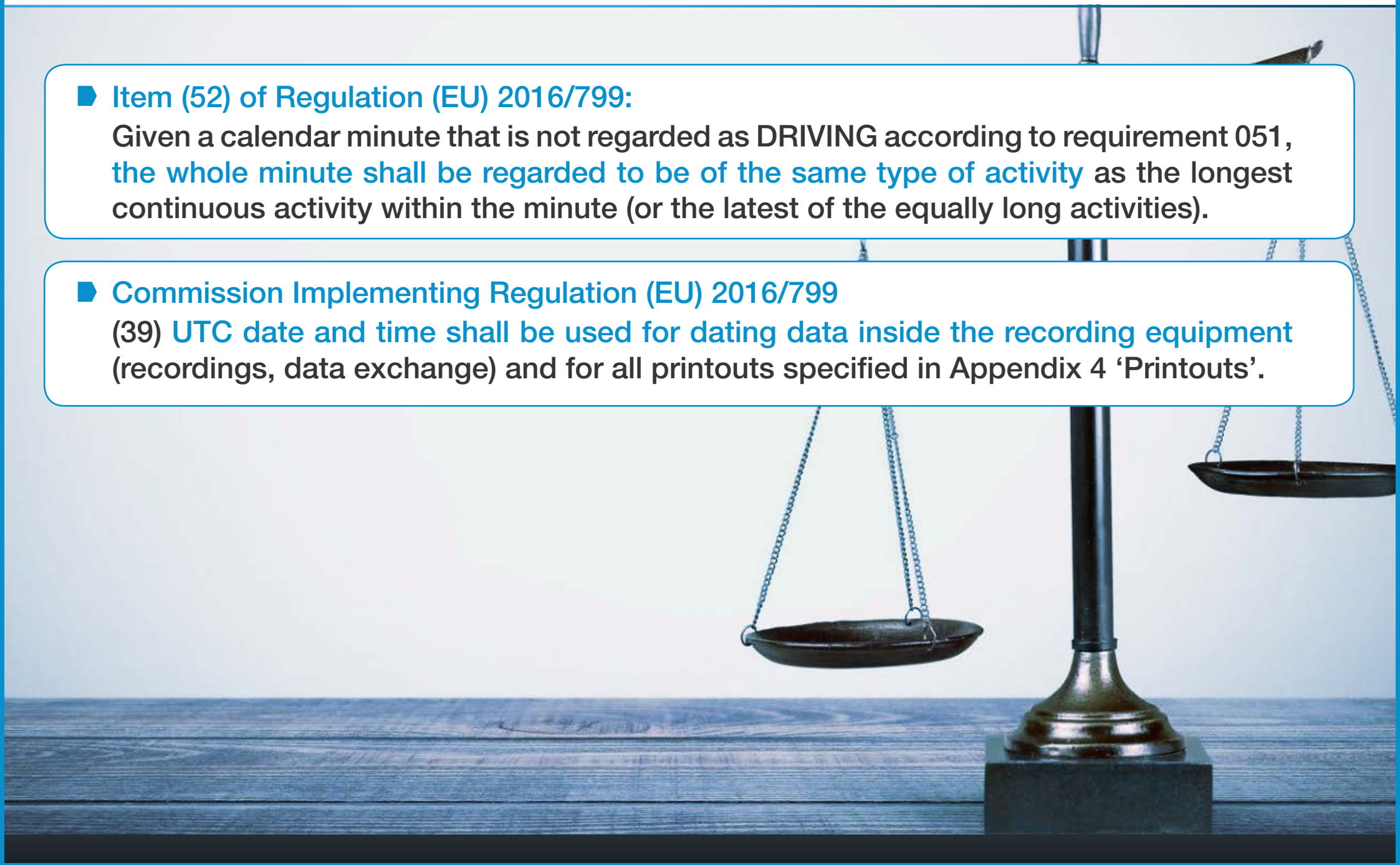
- Item (52) of Regulation (EU) 2016/799:
  Given a calendar minute that is not regarded as DRIVING according to requirement 051, the whole minute shall be regarded to be of the same type of activity as the longest continuous activity within the minute (or the latest of the equally long activities).



UTC AND TACHOGRAPS

- Item (52) of Regulation (EU) 2016/799:
  Given a calendar minute that is not regarded as DRIVING according to requirement 051, the whole minute shall be regarded to be of the same type of activity as the longest continuous activity within the minute (or the latest of the equally long activities).

- Commission Implementing Regulation (EU) 2016/799
  (39) UTC date and time shall be used for dating data inside the recording equipment (recordings, data exchange) and for all printouts specified in Appendix 4 ‘Printouts’.



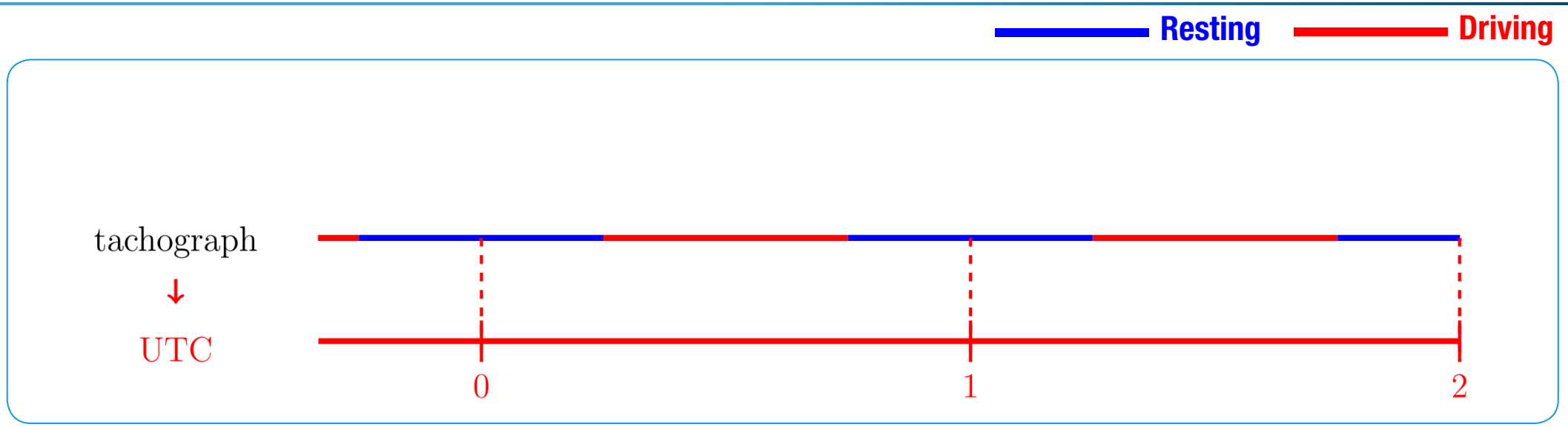
THE BLACK HOLE THEOREM

Resting Driving

tachograph

SUPPOSE THAT THE DRIVER RESTS DURING THE BLUE PERIODS, DRIVES DURING THE RED.

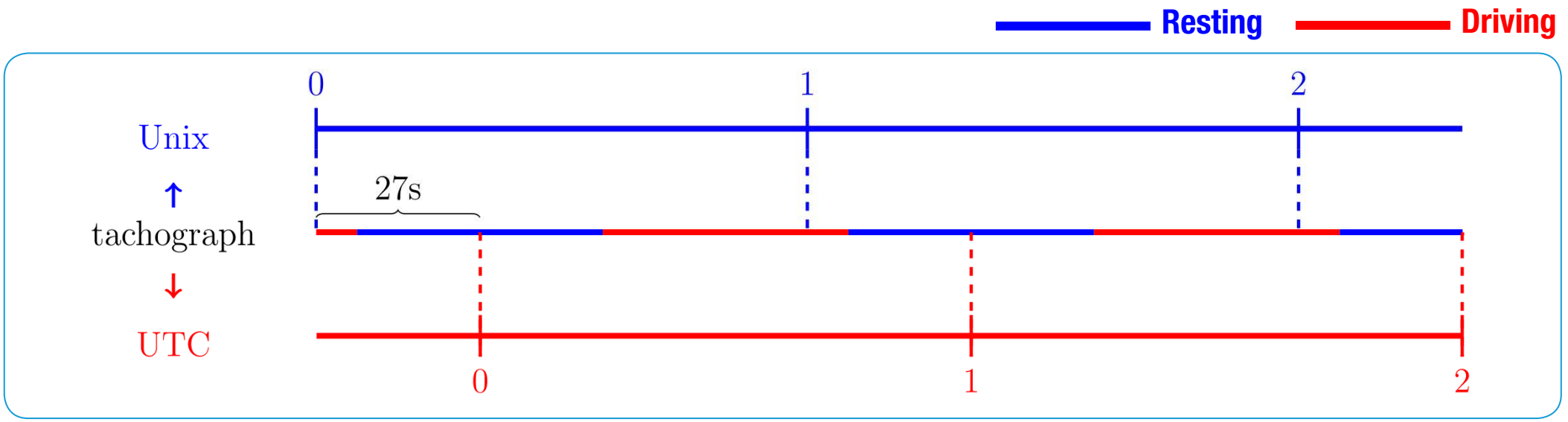
THE BLACK HOLE THEOREM



SUPPOSE THAT THE DRIVER RESTS DURING THE BLUE PERIODS, DRIVES DURING THE RED.

- According to UTC time, the driver would be resting for the whole period.

THE BLACK HOLE THEOREM

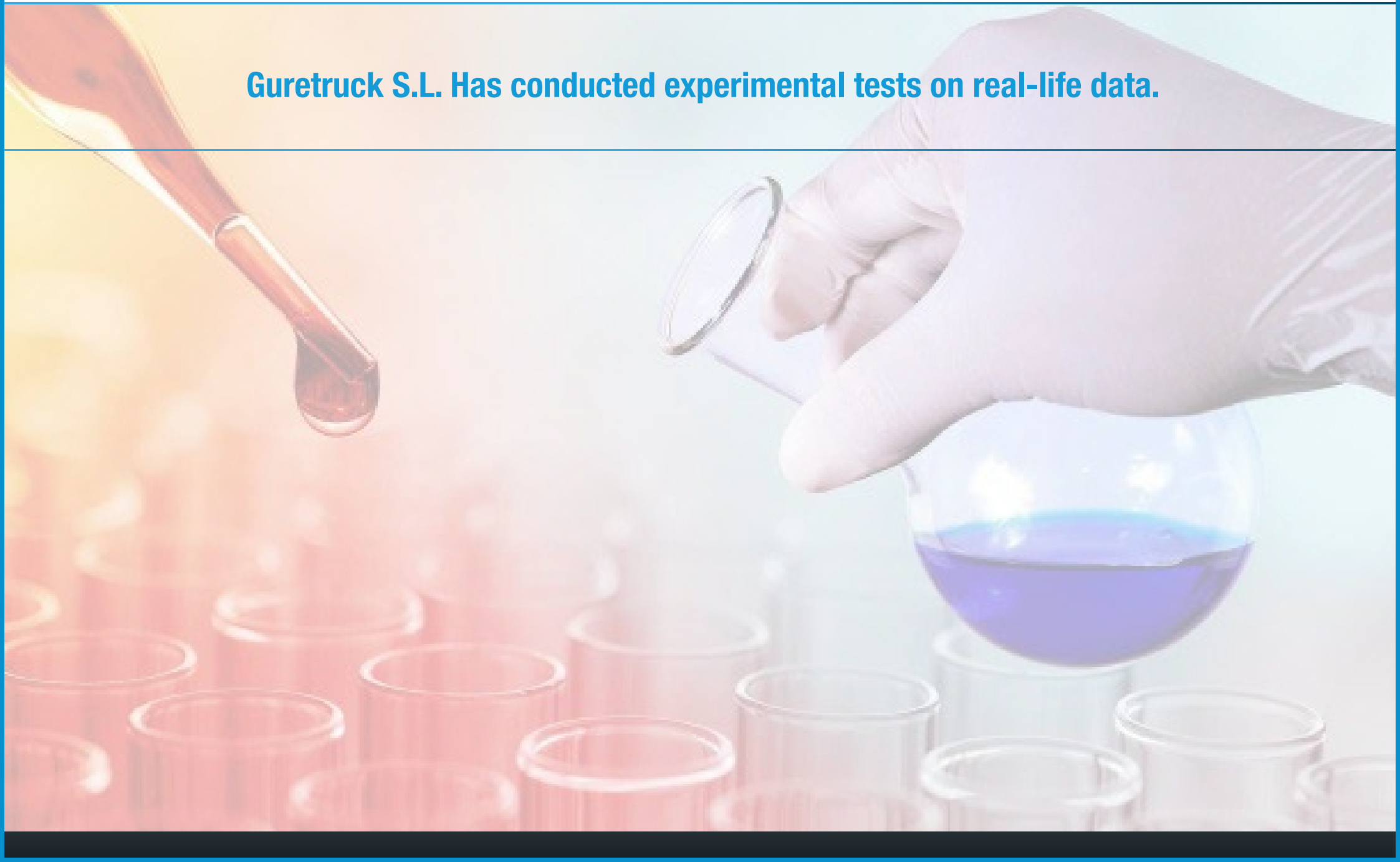


SUPPOSE THAT THE DRIVER RESTS DURING THE BLUE PERIODS, DRIVES DURING THE RED.

- According to UTC time, the driver would be resting for the whole period.
- However, according to Unix time, the driver has been driving the whole time!!

EXPERIMENTAL RESULTS

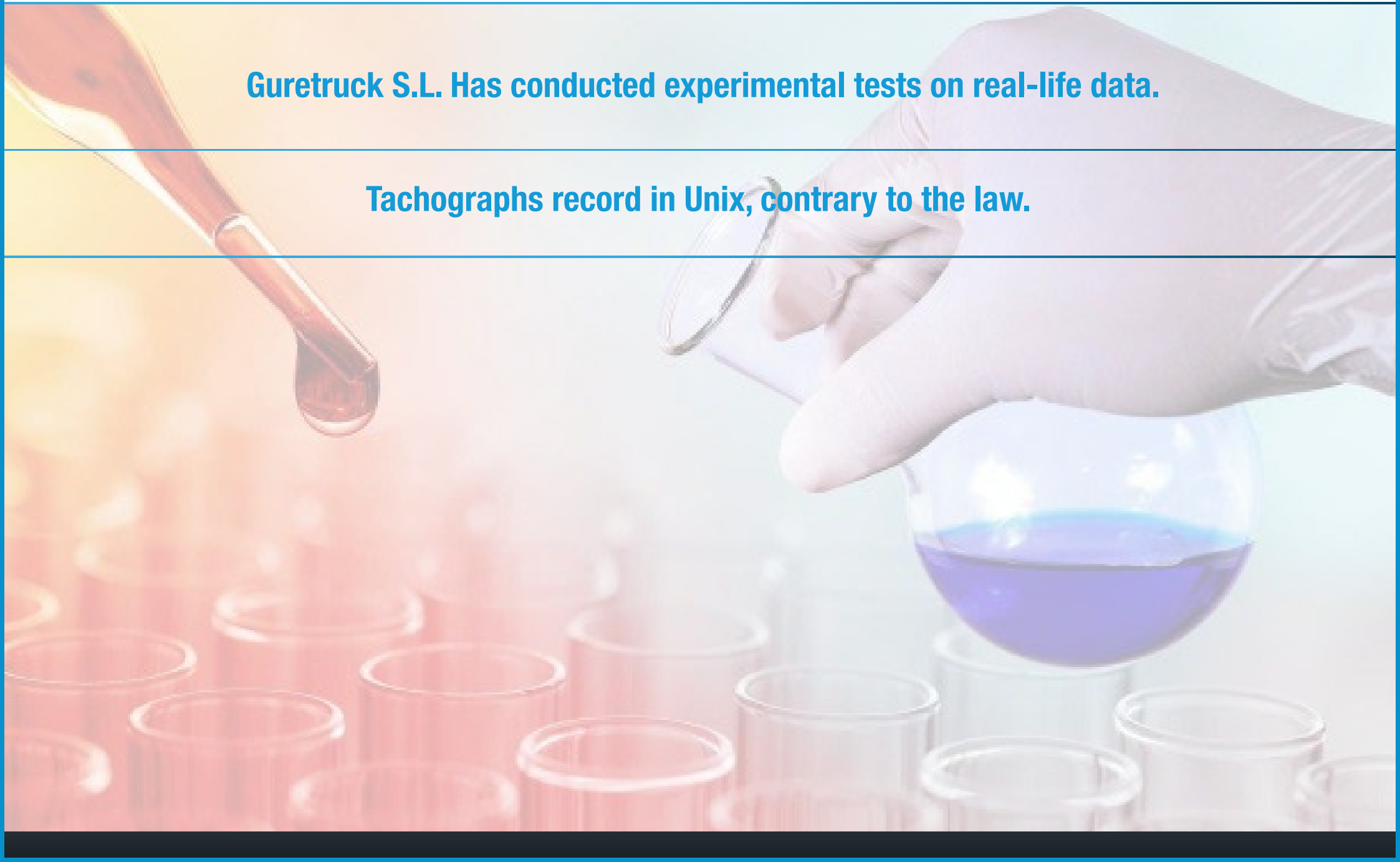
Guretruck S.L. Has conducted experimental tests on real-life data.



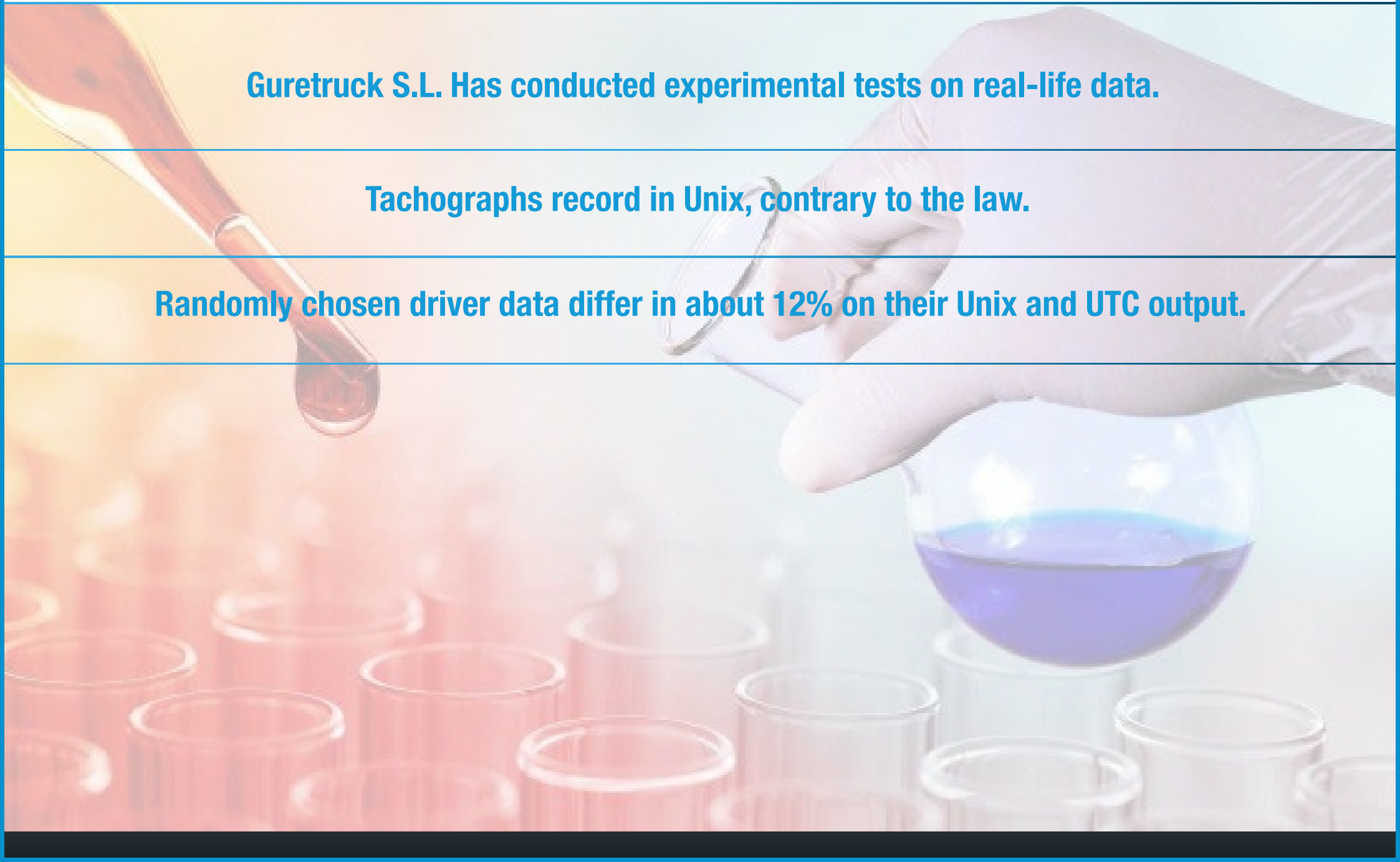
EXPERIMENTAL RESULTS

Guretruck S.L. Has conducted experimental tests on real-life data.

Tachographs record in Unix, contrary to the law.



EXPERIMENTAL RESULTS



Guretruck S.L. Has conducted experimental tests on real-life data.

Tachographs record in Unix, contrary to the law.

Randomly chosen driver data differ in about 12% on their Unix and UTC output.

EXPERIMENTAL RESULTS

Guretruck S.L. Has conducted experimental tests on real-life data.

Tachographs record in Unix, contrary to the law.

Randomly chosen driver data differ in about 12% on their Unix and UTC output.

In theory, these outputs may differ by up to 100%.

EXPERIMENTAL RESULTS

Guretruck S.L. Has conducted experimental tests on real-life data.

Tachographs record in Unix, contrary to the law.

Randomly chosen driver data differ in about 12% on their Unix and UTC output.

In theory, these outputs may differ by up to 100%.

This difference is enough to unjustly send a driver to jail in some countries.

WHAT TO DO?:

Software does not match the regulation, potentially resulting in false convictions.

WHAT TO DO?:

Software does not match the regulation, potentially resulting in false convictions.

- There are at two clear strategies to deal with this:

  (A) VERIFIED SOFTWARE

  (B) AMBIGUITY-FREE LANGUAGE

AMBIGUITY-FREE LANGUAGE

- Formal Vindications is actively working towards developing an ambiguity-free language in which to express the law.

AMBIGUITY-FREE LANGUAGE

- Formal Vindications is actively working towards developing an ambiguity-free language in which to express the law.

- Such a language would be based on computationally efficient logics for reasoning about time, such as Linear Temporal Logic.

AMBIGUITY-FREE LANGUAGE

- Formal Vindications is actively working towards developing an ambiguity-free language in which to express the law.

- Such a language would be based on computationally efficient logics for reasoning about time, such as Linear Temporal Logic.

- In the medium term we expect to develop a language which may be understood by lawmakers and computers alike, greatly automating the process of producing software that matches the legislation.

SCIENTIFIC RESULTS

- Ana de Almeida Borges, Juan José Conejero Rodríguez, David Fernández-Duque, Mireia González Bedmar, Joost J. Joosten:

  The Second Order Traffic Fine: Temporal Reasoning in European Transport Regulations. TIME 2019: 6:1-6:16

SCIENTIFIC RESULTS

- Ana de Almeida Borges, Juan José Conejero Rodríguez, David Fernández-Duque, Mireia González Bedmar, Joost J. Joosten:

  The Second Order Traffic Fine: Temporal Reasoning in European Transport Regulations. TIME 2019: 6:1-6:16

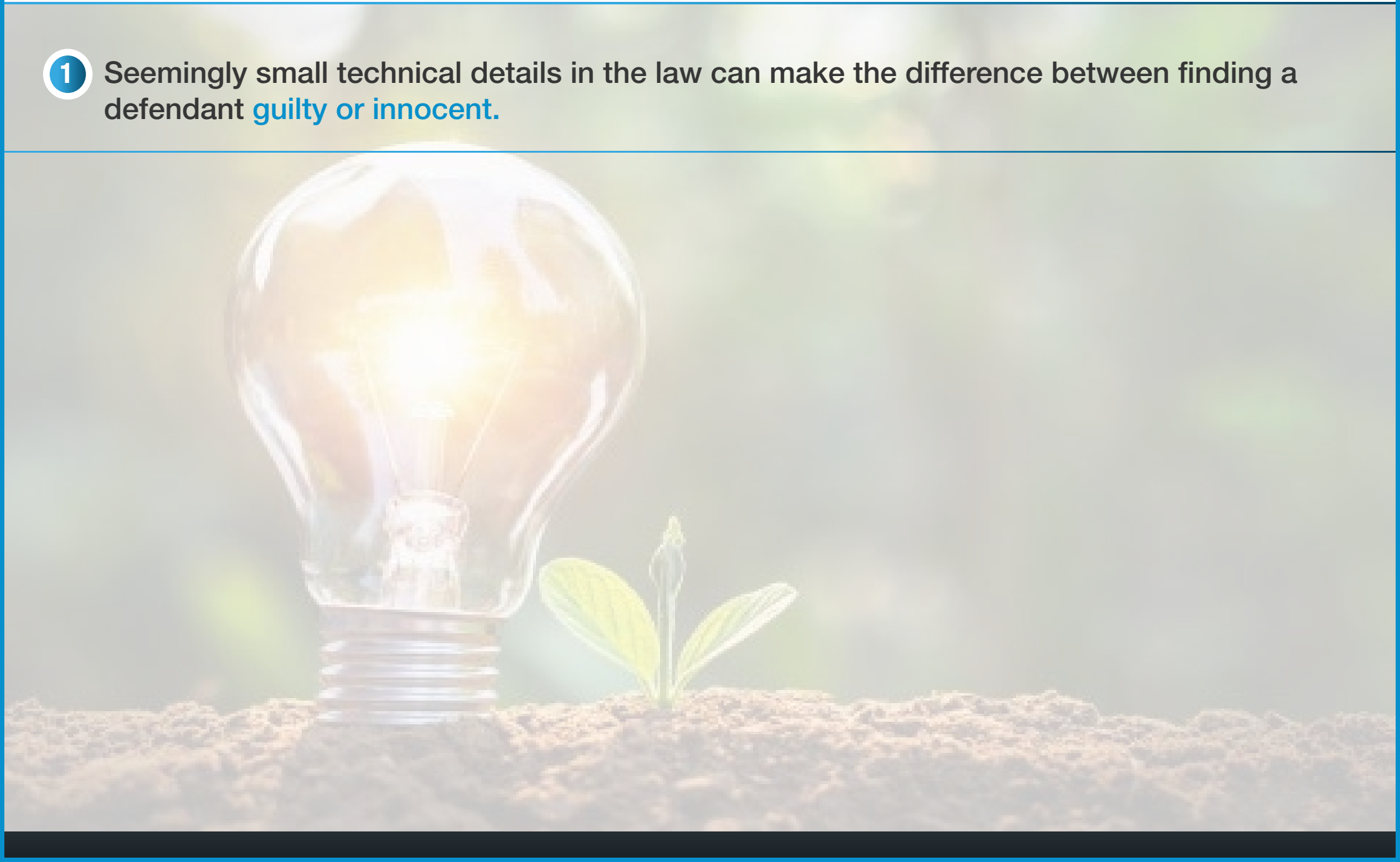
- Ana de Almeida Borges, Juan José Conejero Rodríguez, David Fernández-Duque, Mireia González Bedmar, Joost J. Joosten:

  To Drive or Not to Drive: A Logical and Computational Analysis of European Transport Regulations.

  Accepted for publication in Information and Computation, 2020.

CONCLUSIONS

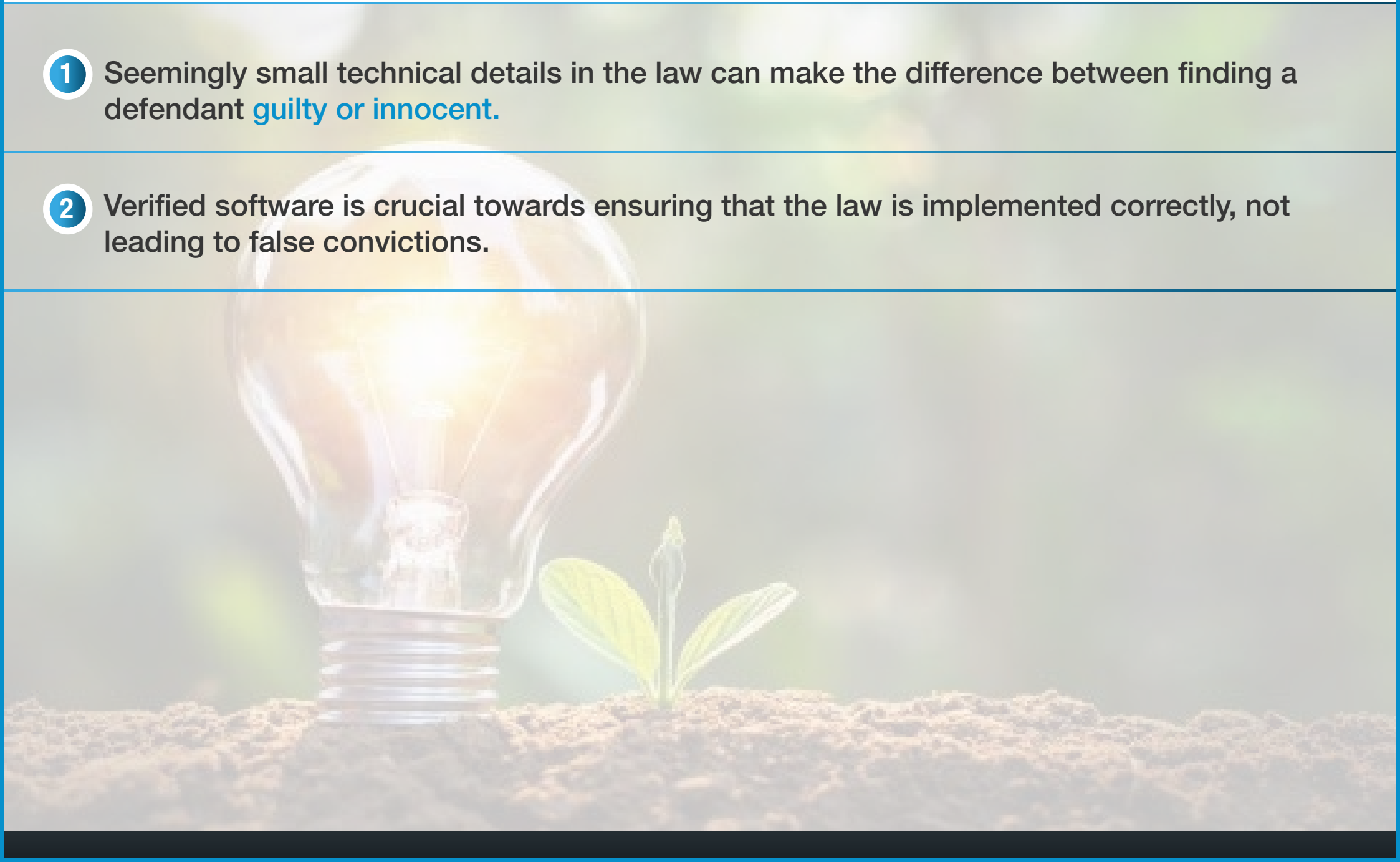
1. Seemingly small technical details in the law can make the difference between finding a defendant guilty or innocent.



CONCLUSIONS

1. Seemingly small technical details in the law can make the difference between finding a defendant guilty or innocent.

2. Verified software is crucial towards ensuring that the law is implemented correctly, not leading to false convictions.

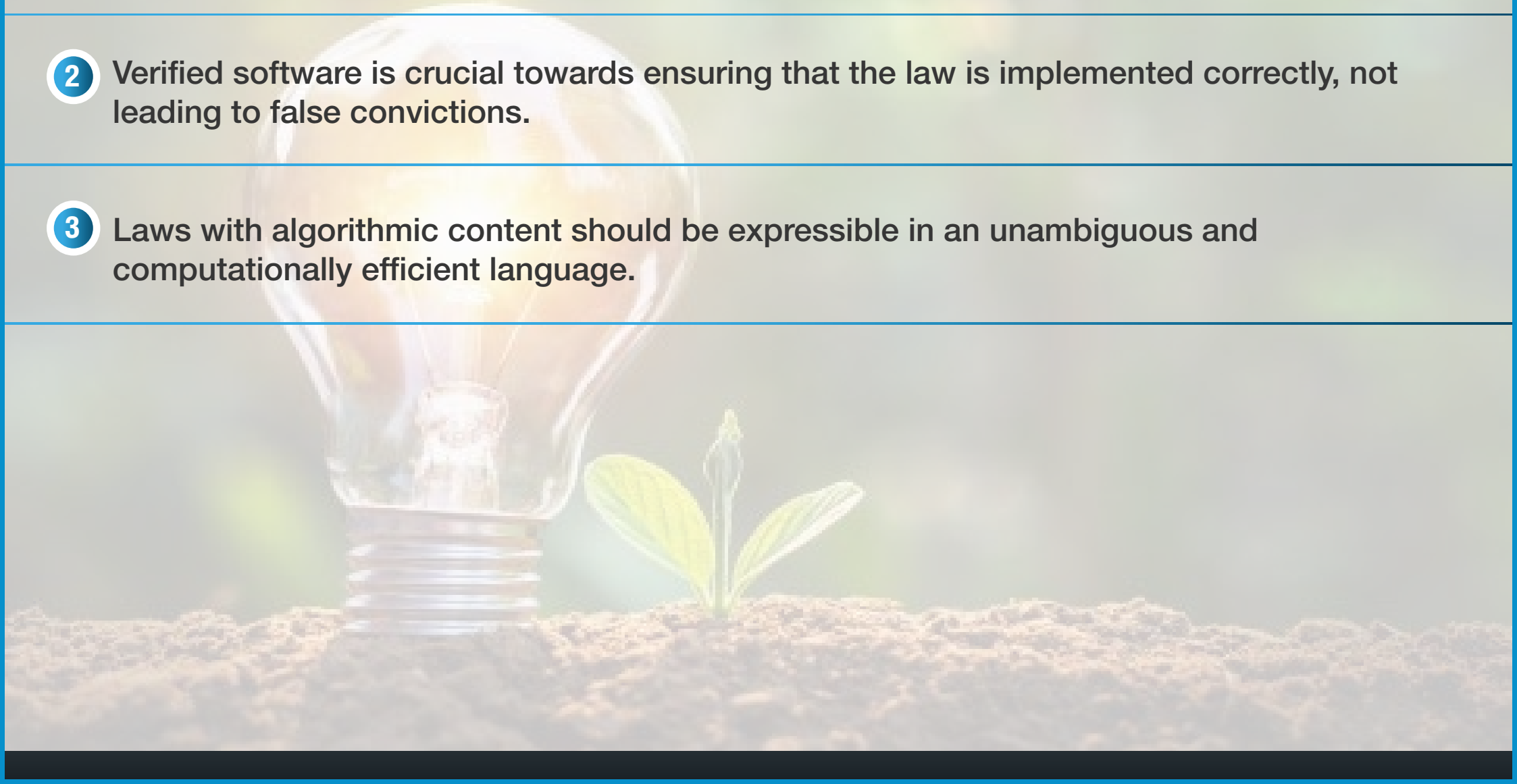


CONCLUSIONS

1. Seemingly small technical details in the law can make the difference between finding a defendant guilty or innocent.

2. Verified software is crucial towards ensuring that the law is implemented correctly, not leading to false convictions.

3. Laws with algorithmic content should be expressible in an unambiguous and computationally efficient language.



CONCLUSIONS

1. Seemingly small technical details in the law can make the difference between finding a defendant guilty or innocent.

2. Verified software is crucial towards ensuring that the law is implemented correctly, not leading to false convictions.

3. Laws with algorithmic content should be expressible in an unambiguous and computationally efficient language.

4. Formal Vindications is at the cutting edge of developing such technology in collaboration with universities.

CONCLUSIONS

1. Seemingly small technical details in the law can make the difference between finding a defendant guilty or innocent.

2. Verified software is crucial towards ensuring that the law is implemented correctly, not leading to false convictions.

3. Laws with algorithmic content should be expressible in an unambiguous and computationally efficient language.

4. Formal Vindications is at the cutting edge of developing such technology in collaboration with universities.

5. The techniques we develop can later be applied to many more legal domains, in any situation where the law describes an algorithm for determining the legality of a given action.

THANK YOU
FOR YOUR ATTENTION!