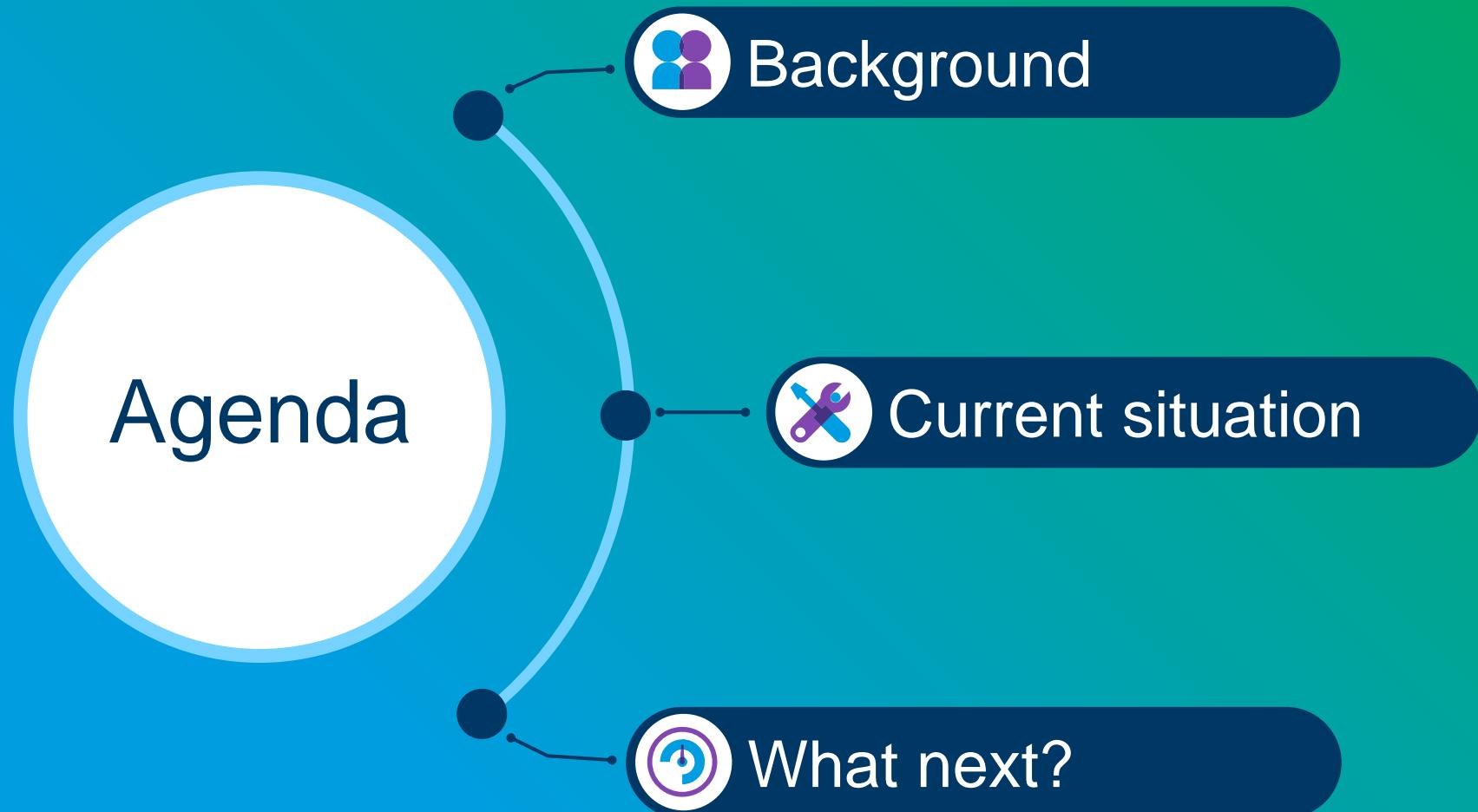


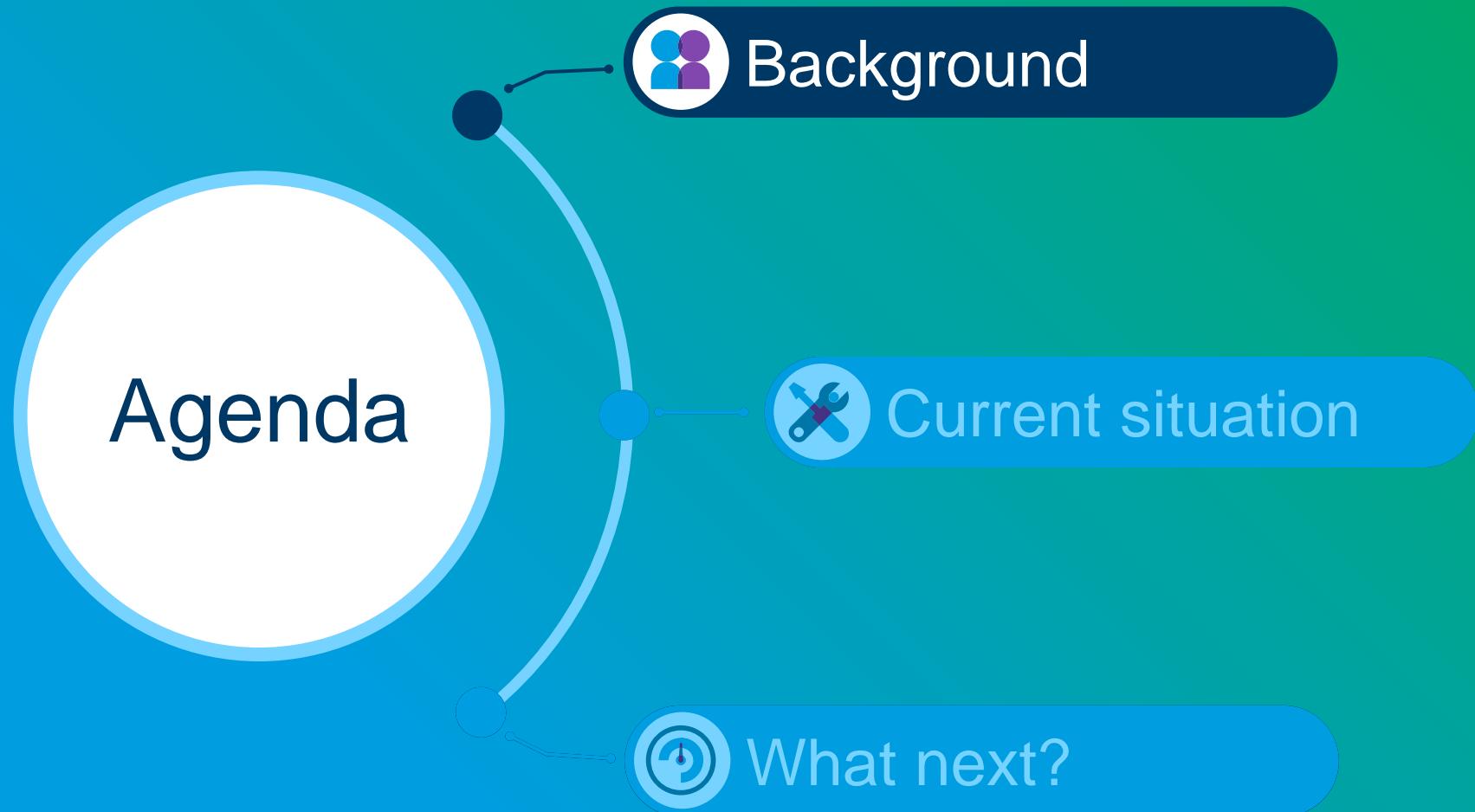
USS valuation

The current situation  
and discussions

DRAFT FOR  
REVIEW  
ONLY







## Contributions

9.8% members

## Current benefits in USS

Retirement Income Builder  
(Defined Benefit)



75ths

£59,883.65

Investment Builder  
(Defined Contribution)



20%

Annual increases

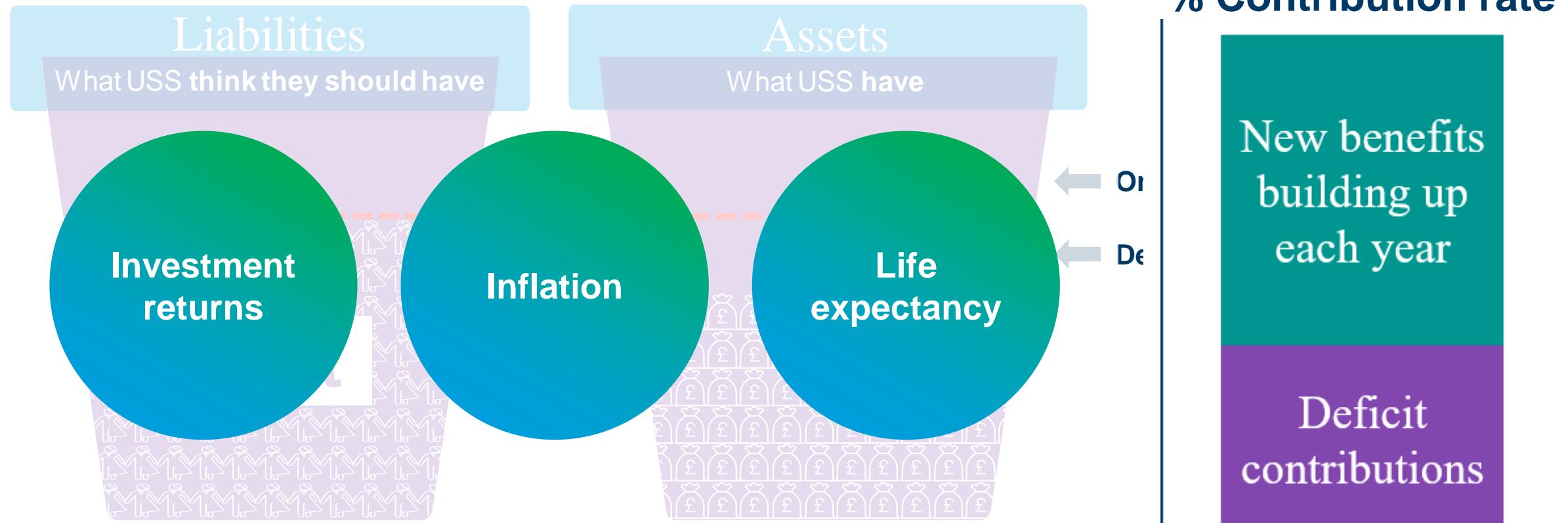
Plus tax free cash  
(3 times pension)

CPI up to certain limits

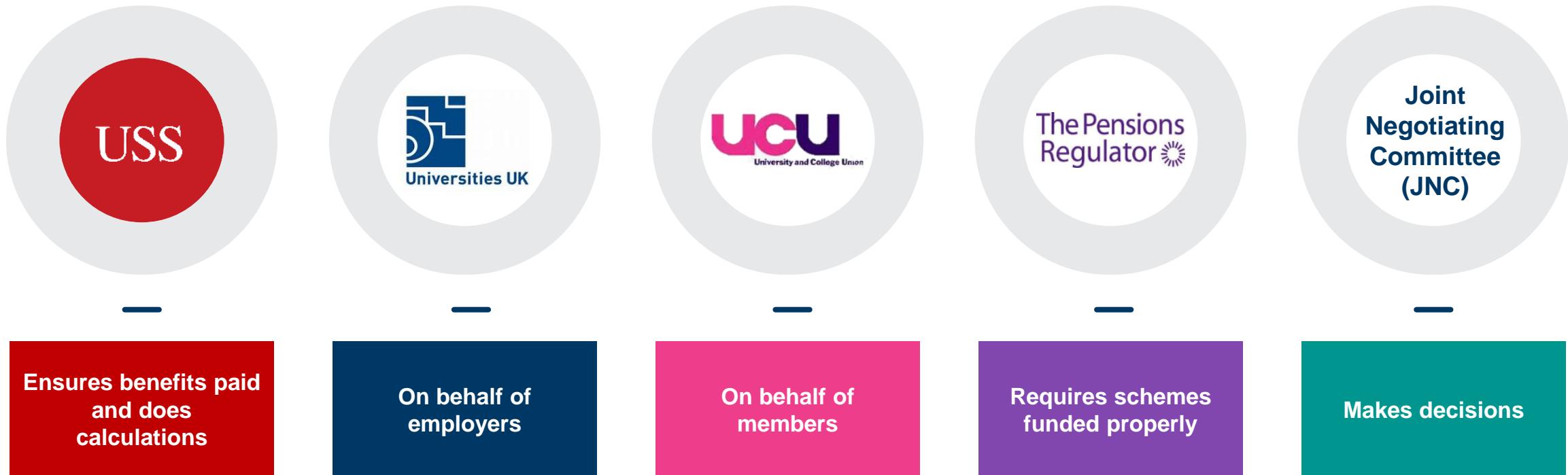
20 40 60 80 100

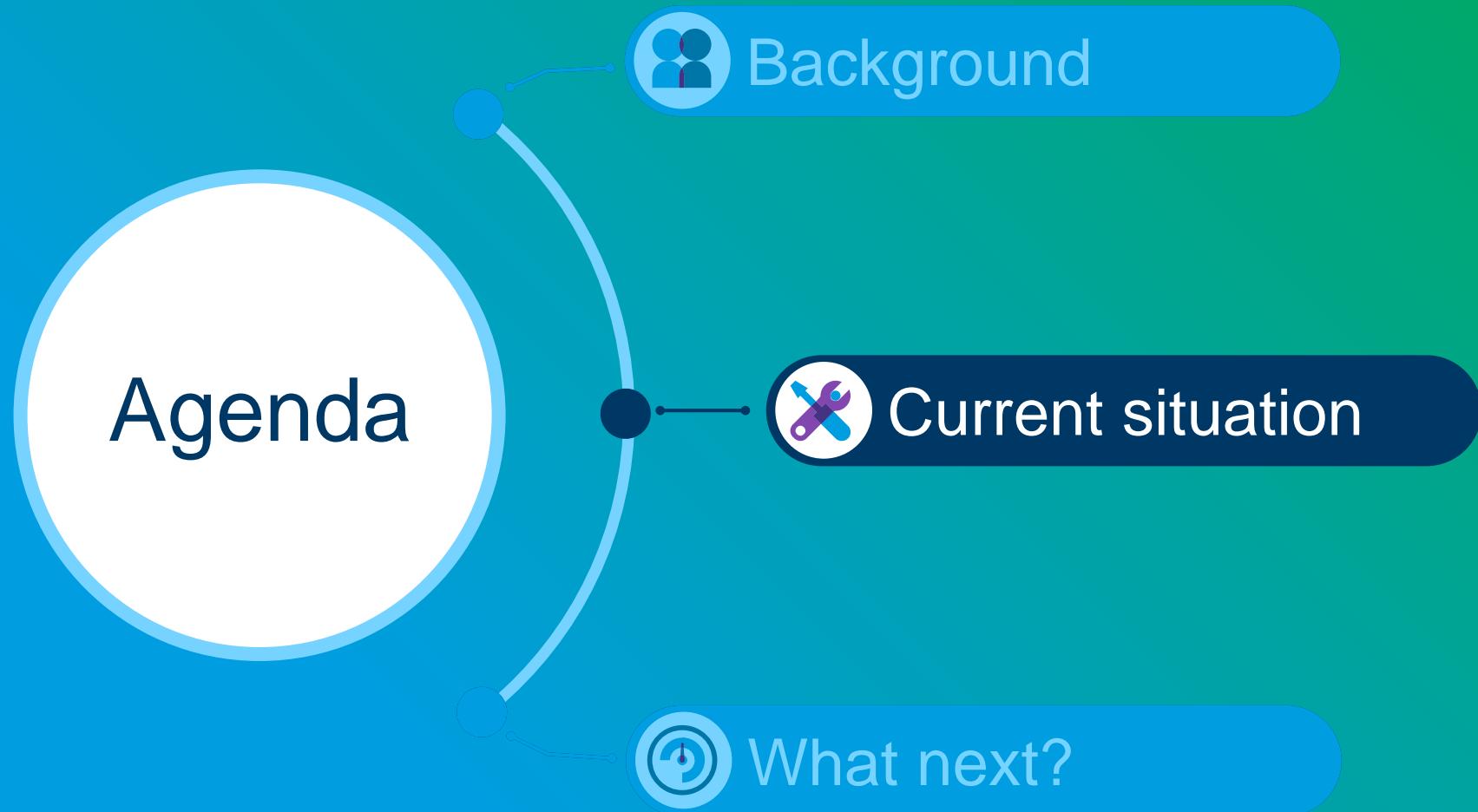
Salary £000s

# Valuation



# Different parties involved





# The 31 March 2020 USS valuation

## Initial results – 3 scenarios

	Contributions	“Adequate” covenant measures	Some covenant measures	No extra covenant measures
Member	9.6%	13.6%	16.3%	18.6%
Employer	21.1%	28.5%	33.3%	37.6%
Total	<b>30.7%</b>	<b>42.1%</b>	<b>49.6%</b>	<b>56.2%</b>

## Covenant enhancing measures

# Strength of the employers?

Debt monitoring

Triggers  
(and pari passu)

Restriction on  
exiting USS



# UUK employer consultation

Benefit changes suggested to keep current contributions



New covenant enhancing measures proposed



Lower cost, flexible benefit options



Governance review of scheme



UUK continued to challenge USS



```
package com.ds.uuc.be.becore.solr;

import ...

public final class LocationUtils {

    /**
     * Parses Point from it's String representation.
     * @param locationString - String that represents location, as 2 double values split with comma
     * @return org.springframework.data.solr.core.geo.Point instance
     */
    public static Point parseLocation(String locationString) {
        Preconditions.checkNotNull(locationString, errorMessage: "Location String should not be null");
        Preconditions.checkNotNull(locationString.contains(","), errorMessage: "Location must be split by comma");
        locationString = locationString.trim();

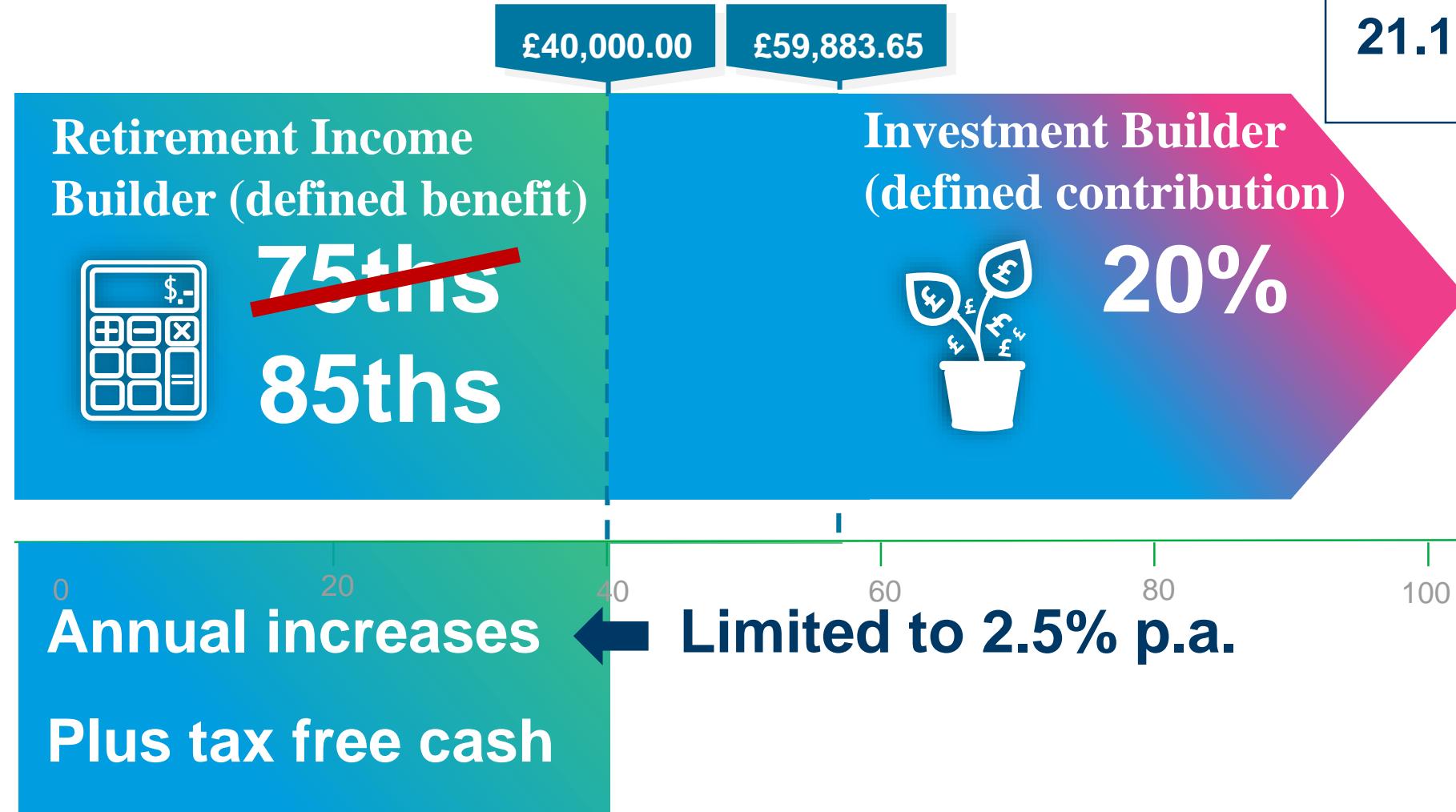
        if (locationString.contains(" ")) {
            locationString = locationString.replaceAll(regex: " ", replacement: ",");
        }

        if (locationString.contains(", ")) {
            locationString = locationString.replaceAll(regex: ", ", replacement: ","));
        }

        String[] location = locationString.split(regex: ",");
        Preconditions.checkNotNull(location, errorMessage: "Location should contain at least 2 values");
        double lat = Double.parseDouble(location[0]);
        double lon = Double.parseDouble(location[1]);

        return new Point(lat, lon);
    }
}
```

# UUK proposal



# USS response to UUK – 9 June 2021

Covenant enhancing measures not strong enough

Can get to **31.2%** with adjusted measures though  
(i.e. 0.5% more than current contributions of 30.7%)

Yes to looking at low cost / flexible options

```
package com.ds.ucd.be.becore.solr;

import ...

public final class LocationUtils {

    /**
     * Parses Point from it's String representation.
     * @param locationString - String that represents location, as 2 double values split with comma
     * @return org.springframework.data.solr.core.geo.Point instance
     */
    public static Point parseLocation(String locationString) {
        Preconditions.checkNotNull(locationString, errorMessage: "Location String should not be null");
        Preconditions.checkNotNull(locationString.contains(","), errorMessage: "Location must be split by comma");
        locationString = locationString.trim();

        if (locationString.contains(" ,")) {
            locationString = locationString.replaceAll(regex: " ,", replacement: " "));
        }

        if (locationString.contains(" ,")) {
            locationString = locationString.replaceAll(regex: " ,", replacement: ","));
        }

        String[] location = locationString.split(regex: ",");
        Preconditions.checkNotNull(location, errorMessage: "Location should contain 2 values");
        double lat = Double.parseDouble(location[0]);
        double lon = Double.parseDouble(location[1]);

        return new Point(lat, lon);
    }
}
```

USS

# TPR get involved – 11 June 2021

Not comfortable with 31.2% but will not take action

See it is a marginal situation

Expect more “normal” assumptions next time

```
package com.ds.ucd.be.becore.solr;

import ...

public final class LocationUtils {

    /**
     * Parses Point from it's String representation.
     * @param locationString - String that represents location, as 2 double values split with comma
     * @return org.springframework.data.solr.core.geo.Point instance
     */
    public static Point parseLocation(String locationString) {
        Preconditions.checkNotNull(locationString, errorMessage: "Location String should not be null");
        Preconditions.checkNotNull(locationString.contains(","), errorMessage: "Location must be split by comma");
        locationString = locationString.trim();

        if (locationString.contains(" ,")) {
            locationString = locationString.replaceAll(regex: " ,", replacement: " "));
        }

        if (locationString.contains(" ,")) {
            locationString = locationString.replaceAll(regex: " , ", replacement: " ,"));
        }

        String[] location = locationString.split(regex: ", ");
        Preconditions.checkNotNull(location, errorMessage: "Location should contain 2 values");
        double lat = Double.parseDouble(location[0]);
        double lon = Double.parseDouble(location[1]);

        return new Point(lat, lon);
    }
}
```

The Pensions  
Regulator

# UUK's next consultation – 18 June 2021

## A further short consultation with employers

### Benefits



### Covenant measures



### Contributions

**9.8% members  
21.4% employers**

**0.5% more**

UUK ask if the 0.5% can be removed

UUK asks about a 31 March 2021 valuation

# USS response to UUK – 28 June 2021

31.2% not moving unless change something else

At the very boundary of USS risk appetite

31 March 2021 unlikely to be any different  
but they will look at properly

```
package com.ds.ucd.be.becore.solr;

import ...

public final class LocationUtils {

    /**
     * Parses Point from it's String representation.
     * @param locationString - String that represents location, as 2 double values split with comma
     * @return org.springframework.data.solr.core.geo.Point instance
     */
    public static Point parseLocation(String locationString) {
        Preconditions.checkNotNull(locationString, errorMessage: "Location String should not be null");
        Preconditions.checkNotNull(locationString.contains(","), errorMessage: "Location must be split by comma");
        locationString = locationString.trim();

        if (locationString.contains(" ,")) {
            locationString = locationString.replaceAll(regex: " ,", replacement: " "));
        }

        if (locationString.contains(" ,")) {
            locationString = locationString.replaceAll(regex: " ,", replacement: " ,"));
        }

        String[] location = locationString.split(regex: ",");
        Preconditions.checkNotNull(location, errorMessage: "Location should contain 2 values");
        double lat = Double.parseDouble(location[0]);
        double lon = Double.parseDouble(location[1]);

        return new Point(lat, lon);
    }
}
```

USS

# TPR comment on a 2021 valuation – 14 July 2021

Expect 2020 valuation to be finalised without delay



Would expect more “normal” parameters at 2021



```
package com.ds.ucd.be.becore.solr;

import ...

public final class LocationUtils {

    /**
     * Parses Point from it's String representation.
     * @param locationString - String that represents location, as 2 double values split with comma
     * @return org.springframework.data.solr.core.geo.Point instance
     */
    public static Point parseLocation(String locationString) {
        Preconditions.checkNotNull(locationString, errorMessage: "Location String should not be null");
        Preconditions.checkNotNull(locationString.contains(","), errorMessage: "Location must be split by comma");
        locationString = locationString.trim();

        if (locationString.contains(" ,")) {
            locationString = locationString.replaceAll(regex: ", ", replacement: " ");
        }

        if (locationString.contains(" ,")) {
            locationString = locationString.replaceAll(regex: ", ", replacement: ","));
        }

        String[] location = locationString.split(regex: ",");
        Preconditions.checkNotNull(location.length == 2, errorMessage: "Location should contain 2 values");
        double lat = Double.parseDouble(location[0]);
        double lon = Double.parseDouble(location[1]);

        return new Point(lat, lon);
    }
}
```

The Pensions  
Regulator A circular logo consisting of a series of small, light blue and white triangles pointing outwards, forming a sunburst or gear-like pattern.

# USS briefing on a 2021 valuation – 23 July 2021

CURRENT BENEFITS	31 March 2020	31 March 2021
<b>Future service cost</b>	33.6%	36.7%
<b>Deficit</b>	£15.4bn	£7.1bn
<b>Deficit contributions</b>	8.9%	6%
<b>Total contributions</b>	42.5%	42.7%

# JNC decision – 31 August 2021

JNC have voted to progress the UUK proposal for benefits from April 2022

Casting vote of the Chair needed but UCU proposal not formally tabled to be voted on

Short service and low cost options to be considered

Member consultation in the autumn

Joint Negotiating Committee (JNC)

```
package com.ds.ucd.be.bcore.solr;

import ...

public final class LocationUtils {

    /**
     * Parses Point from it's String representation.
     * @param locationString - String that represents location, as 2 double values split with comma
     * @return org.springframework.data.solr.core.geo.Point instance
     */
    public static Point parseLocation(String locationString) {
        Preconditions.checkNotNull(locationString, errorMessage: "Location String should not be null");
        Preconditions.checkNotNull(locationString.contains(","), errorMessage: "Location must be split by comma");
        locationString = locationString.trim();

        if (locationString.contains(",,")) {
            locationString = locationString.replaceAll(regex: ",,", replacement: " ");
        }

        if (locationString.contains(", ")) {
            locationString = locationString.replaceAll(regex: ", ", replacement: ",");

        }

        String[] location = locationString.split(regex: ",");
        Preconditions.checkNotNull(location, errorMessage: "Location should contain at least 2 values");
        double lat = Double.parseDouble(location[0]);
        double lon = Double.parseDouble(location[1]);

        return new Point(lat, lon);
    }
}
```

# UCU's press release on 31 August 2021

UCU is discussing action with its members

UCU's proposal: £40,000 threshold, 80ths accrual,  
no 2.5% cap on increases

Contributions of 8.1% and 24.9%

```
package com.ds.ucd.be.bcore.solr;

import ...

public final class LocationUtils {

    /**
     * Parses Point from it's String representation.
     * @param locationString - String that represents location, as 2 double values split with comma
     * @return org.springframework.data.solr.core.geo.Point instance
     */
    public static Point parseLocation(String locationString) {
        Preconditions.checkNotNull(locationString, errorMessage: "Location String should not be null");
        Preconditions.checkNotNull(locationString.contains(","), errorMessage: "Location must be split by comma");
        locationString = locationString.trim();

        if (locationString.contains(", ")) {
            locationString = locationString.replaceAll(regex: ", ", replacement: " ");
        }

        if (locationString.contains(",")) {
            locationString = locationString.replaceAll(regex: ", ", replacement: ","));
        }

        String[] location = locationString.split(regex: ",");
        Preconditions.checkArgument(location.length == 2, errorMessage: "Location should contain 2 values");
        double lat = Double.parseDouble(location[0]);
        double lon = Double.parseDouble(location[1]);

        return new Point(lat, lon);
    }
}
```



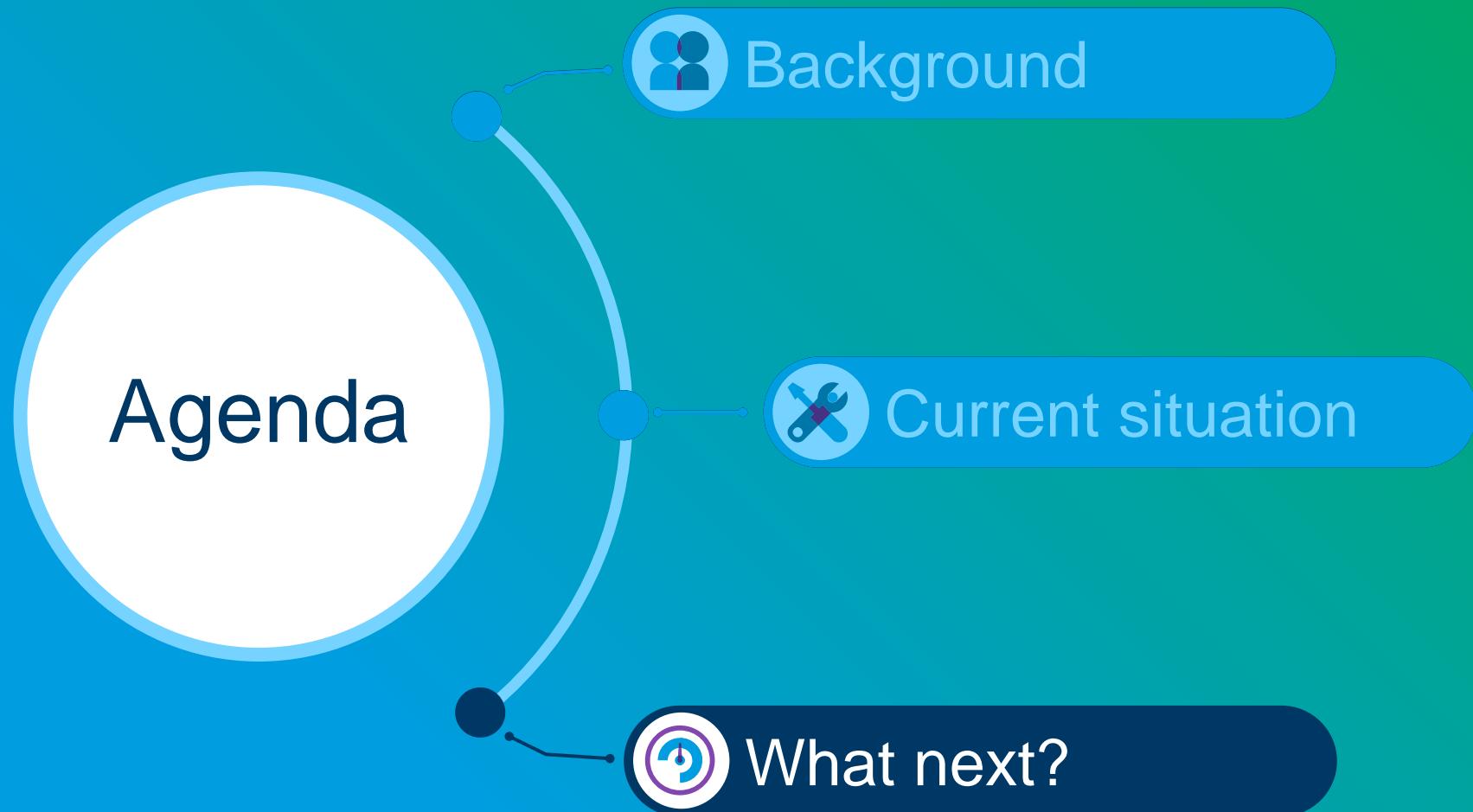
# The 1 October 2021 contributions

New contributions of **9.8%** and **21.4%** from October 2021  
(so they didn't go to 11% and 23.7% as scheduled)

BUT, USS need “back-stop” position if reforms don’t come in

Back-stop	Members	Employers
April 2022	<b>11.0%</b>	<b>23.7%</b>
October 2022	<b>12.9%</b>	<b>27.1%</b>
April 2023	<b>13.9%</b>	<b>29.1%</b>
October 2023	<b>15.0%</b>	<b>31.0%</b>
April 2024	<b>16.0%</b>	<b>33.0%</b>
October 2024	<b>17.1%</b>	<b>34.9%</b>
April 2025	<b>18.1%</b>	<b>36.9%</b>
October 2025	<b>18.8%</b>	<b>38.2%</b>





## What next?



**Have your say!**  
Ask questions  
Give feedback



Information  
booklets

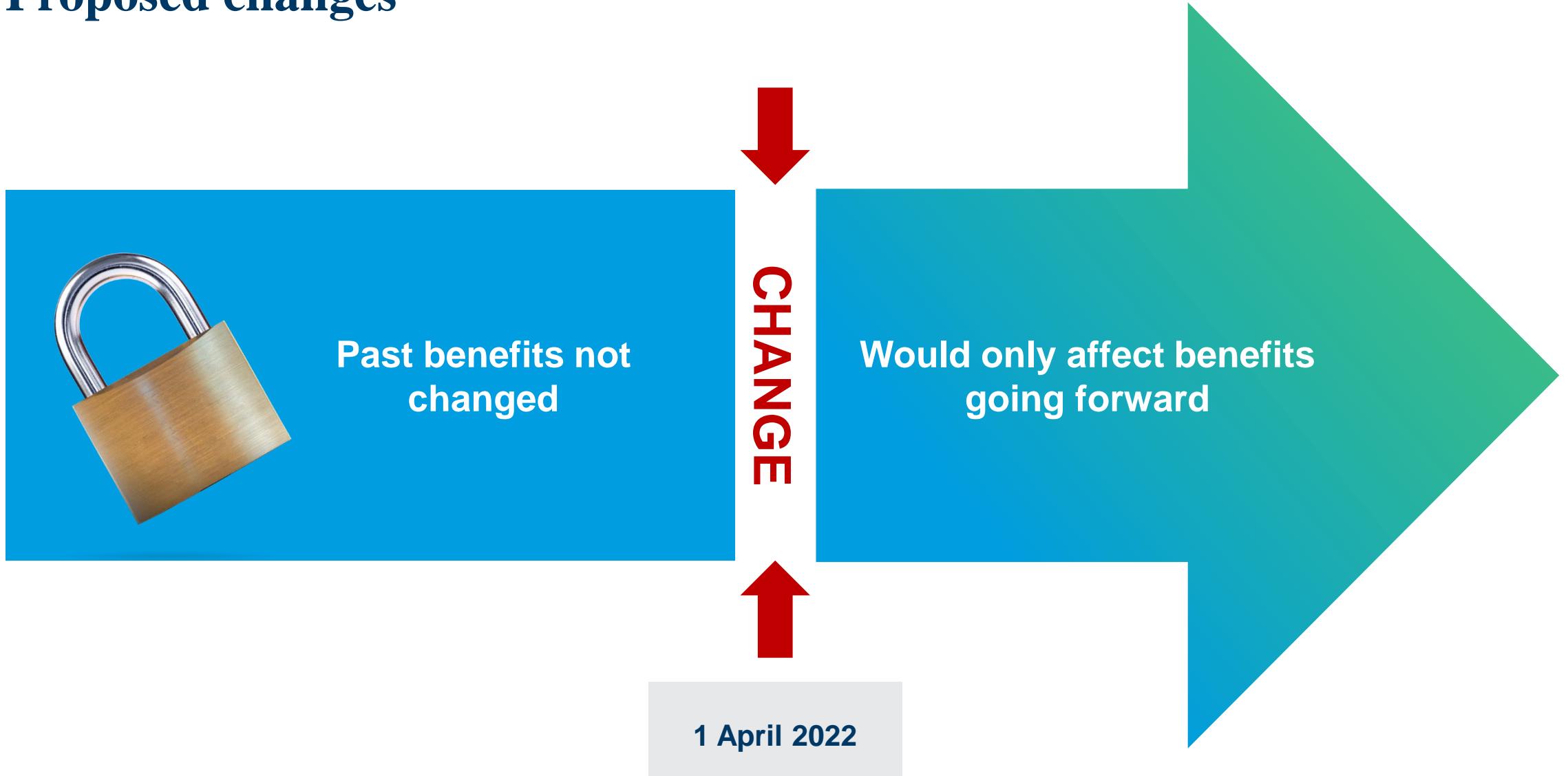
Website

Modellers

November 2021

1 April 2022

# Proposed changes



## Contributions

**9.8% members**

## UUK proposal

