Supporting Information

Self-regulating and self-oscillating Metal-Organic Framework based plasmonic metasurfaces

Hajar Amyar,[1,2] Davide Raffaele Ceratti,[1] Henri Benisty,[2] Andrea Cattoni,[3] Mondher Besbes,[2] Marco Faustini[1,4]*

1 Sorbonne Université, CNRS, Laboratoire Chimie de la Matière Condensée de Paris (LCMCP), F-75005 Paris, France

2 Université Paris-Saclay, Institut d'Optique Graduate School, CNRS, Laboratoire Charles Fabry, 91127 Palaiseau, France

3 Centre de Nanosciences et de Nanotechnologies (C2N), CNRS UMR 9001, Université Paris-Saclay, Palaiseau, France.

4 Institut Universitaire de France (IUF), 75231 Paris, France

E-mail: marco.faustini@sorbonne-universite.fr

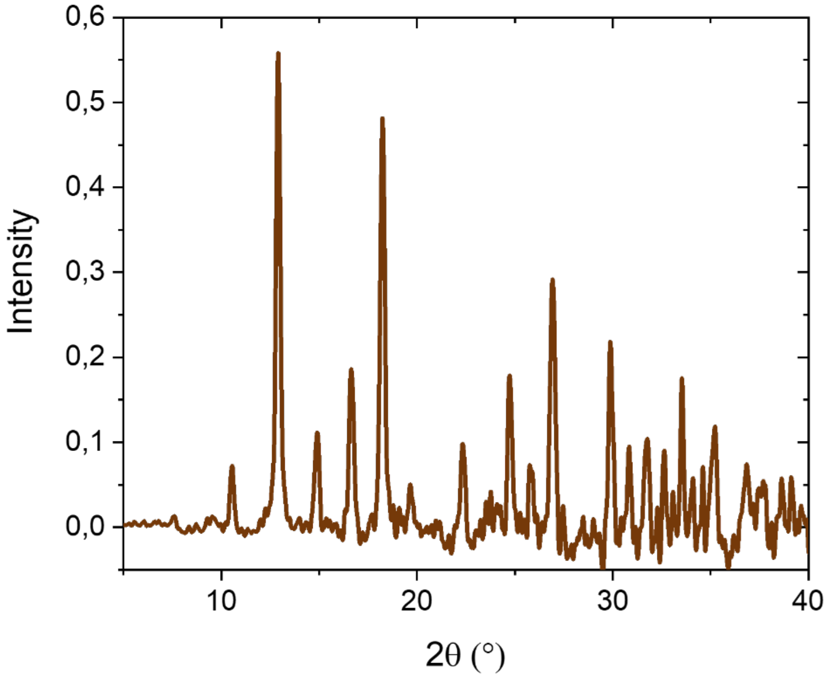

Figure S1 XRD pattern of the ZIF-8 colloidal film.

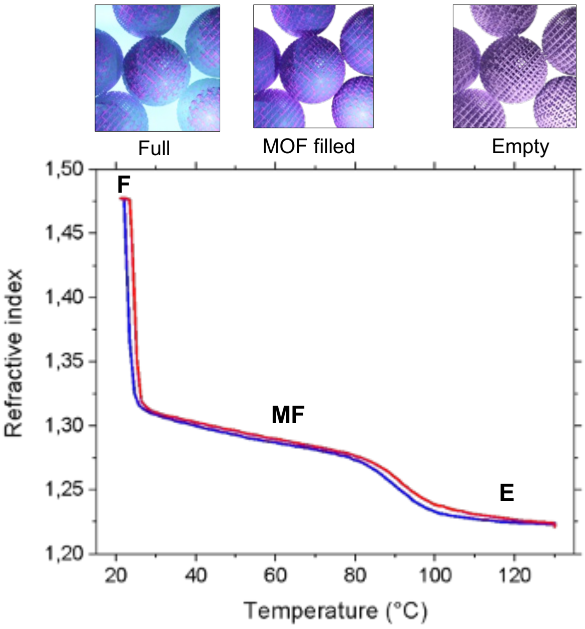

Figure S2 Refractive index as function of the temperature of the ZIF-8 layer in presence of 0.7 P/P_0 of IPA.

Calculation of the porous volume

Starting from the values of refractive index associated to the three states E, MF and F we can evaluate the porous volume by using the Bruggemann effective medium approximation (BEMA) model (eq 1).[1] First, by taking the dielectric constant ε of the film in the empty (E) and filled state (F), this model allows the estimation of the dielectric constant of solid wall of the ZIF-8 material (ε_{MOF}). In the empty and full states the BEMA model can be written as follows:

$$f_{MOF}\frac{\tilde{\varepsilon}_{MOF}-\tilde{\varepsilon}}{\tilde{\varepsilon}_{MOF}+2\tilde{\varepsilon}}+f_A\frac{\tilde{\varepsilon}_A-\tilde{\varepsilon}}{\tilde{\varepsilon}_A+2\tilde{\varepsilon}}=0 \qquad \text{E state} \qquad (1)$$

$$f_{MOF}\frac{\tilde{\varepsilon}_{MOF}-\tilde{\varepsilon}}{\tilde{\varepsilon}_{MOF}+2\tilde{\varepsilon}}+f_W\frac{\tilde{\varepsilon}_{iso}-\tilde{\varepsilon}}{\tilde{\varepsilon}_{iso}+2\tilde{\varepsilon}}=0 \quad \text{F state} \qquad (2)$$

Where f_{MOF} , f_w and f_A and ε_{MOF}, ε_{iso}, ε_A the relative volumetric fractions and dielectric constants of three compounds MOF, Iso (Isopropanol) and A (Air). We consider the dielectric constants to be the square of refractive index values (real part of the dielectric constants). This assumption can be made for media that not absorb light in the considered range of wavelengths (typically 400-1000nm). Cauchy model for "transparent" materials was employed to fit ZIF-8 film.[2]

According to the scheme in Figure (2b), in the Filled state (F) all the air that was initially present in the film (porosity of the MOF + interparticle void) is replaced with isopropanol vapor. We can thus assume that:

$F_{Iso} = f_A$ (3)

that also corresponds to the total porous volume. According to Figure (2b), the refractive indices of the empty and filled states are 1.22 and 1.47 respectively. Considering the refractive indices of isopropanol and air equal to 1.37 and 1 respectively, the porous volume and the ε_{MOF} (and the refractive index) can be calculated by combining equations (1), (2) and (3).

Following this calculation, the refractive index of the ZIF-8 wall is deduced to be equal to 1.63 while the total porous volume is 63%. By knowing these values, we can calculate the percentage of porosity associated to each family of mesopores by applying a 3 phase BEMA relation for intemediate states:

$$f_{MOF}\frac{\tilde{\varepsilon}_{MOF}-\tilde{\varepsilon}}{\tilde{\varepsilon}_{MOF}+2\tilde{\varepsilon}}+f_A\frac{\tilde{\varepsilon}_A-\tilde{\varepsilon}}{\tilde{\varepsilon}_A+2\tilde{\varepsilon}}+f_W\frac{\tilde{\varepsilon}_{Iso}-\tilde{\varepsilon}}{\tilde{\varepsilon}_{Iso}+2\tilde{\varepsilon}}=0 \qquad \text{E < state < F} \qquad (4)$$

In which:

$f_{MOF} + f_A + f_{Iso} = 1$ (5)

Since f_{MOF} is equal to 0.37 this relationship can be rewritten as follows

$$F_{Iso} = 0.63 - f_A \qquad (6)$$

and introduced in equation (4). From this calculation, the micropores and interparticle mesoporosities represent around 16% and 47% of the total porosity.

FEM-Based Simulation Methods

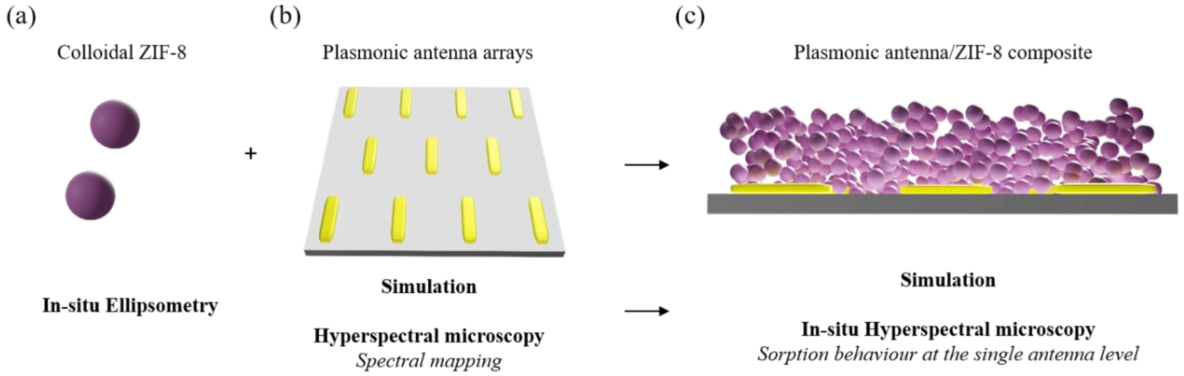

Figure S3 Schematic of the experimental and simulated system using colloidal ZIF-8 nanoparticle and gold nanoantenna. The process involved several steps, including (a) synthesis and characterization of colloidal ZIF-8 nanoparticles with in-situ ellipsometry, (b) simulation and fabrication of gold nanoantenna using e-beam lithography, characterization with hyperspectral microscopy, and (c) deposition of colloidal film onto fabricated nanoantenna. The resulting system was studied with in-situ hyperspectral microscopy, including both experimental measurements and simulations.

We employed the Finite Element Method (FEM) to solve Maxwell's equations that describe the propagation of electromagnetic waves through complex structures. These equations are influenced by the geometry and material properties of the structure. The optical properties of the gold nanoantennas are described using the complex refractive index derived from the Johnson and Christy model.[3] As shown in figure S4 (a) the computational domains were created using CUBIT software, then meshed with triangles and simulated in MATLAB environment. For the ZIF-8 film, the medium is considered homogeneous, with a variable refractive index (n) obtained from the experimental ellipsometry data. We apply a plan wave with E_x polarised and we studied the quarter of the entire structure. We added a perfectly matched layer (PML) in the X, Y and Z directions to enforce boundary condition and absorb outgoing waves without reflecting.

The FEM solution is obtained by transforming the PDEs into a system of algebraic equations. This process involves using finite elements to approximate the solution over the domain. The obtained equations are the Helmholtz equation for the diffracted light. For a three-dimensional simulation with transverse electric (TE) polarization, the FEM formulation with edge elements is given by the following wave equation:

$$\nabla \times \left(\frac{1}{\mu_r} \nabla \times \boldsymbol{E_d}\right) = k_0^2 \epsilon_r \boldsymbol{E}_d + k_{\boldsymbol{0}}^2 (\epsilon_r - \epsilon_b) \boldsymbol{E_{Inc}}$$

Where, $\boldsymbol{E_d}$ is the electric field in the domain, k_0 is the free space wavenumber, ϵ_r is the relative electric permittivity, $\epsilon_b$is the background permittivity and μ_r is the relative magnetic permeability

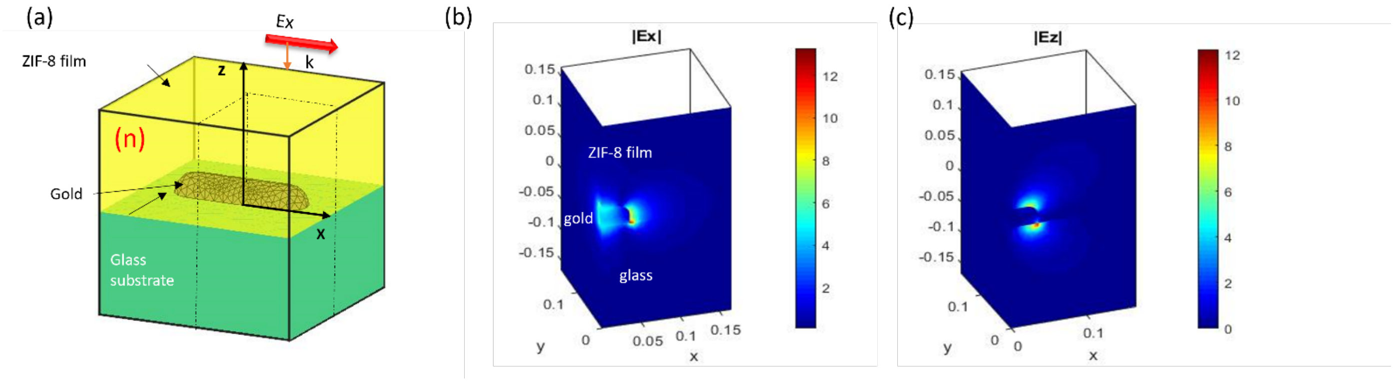

Figure S4 (a) the simulated system with a gold nanoantenna placed on top of a glass substrate, with the nanoantenna surrounded by a homogeneous layer. This layer's refractive index is varied in the simulation to model different experimental conditions for the adsorption/desorption process in zif-8 film with isopropanol vapor. (b,c) the distribution of the electric field of the gold nanoantenna along the x and z components, respectively.

By following this weak formulation and discretizing the problem using edge elements, we can obtain the electric field distribution in the structure (Figure (S4 b-c) along the x,y and z component for single antenna. The scattering cross-section is determined based on the calculations of the extinction and absorption cross-sections as follows:

$$\sigma_{Scat} = \sigma_{Ext} - \sigma_{Abs}$$

The absorptivity (A) is calculated as the ratio of the absorbed power $P_{Ads}$to the incident power P_{Inc}.The absorbed power is obtained by summing the absorption cross-sections $\sigma_{Abs}$of individual antennas and multiplying by the incident power density σ_{Abs}. The formula for absorptivity is:

$$A = \frac{P_{Ads}}{P_{Inc}} = \frac{(\Sigma\ \sigma_{Abs}\ *\ I_{Inc})}{P_{Inc}}$$

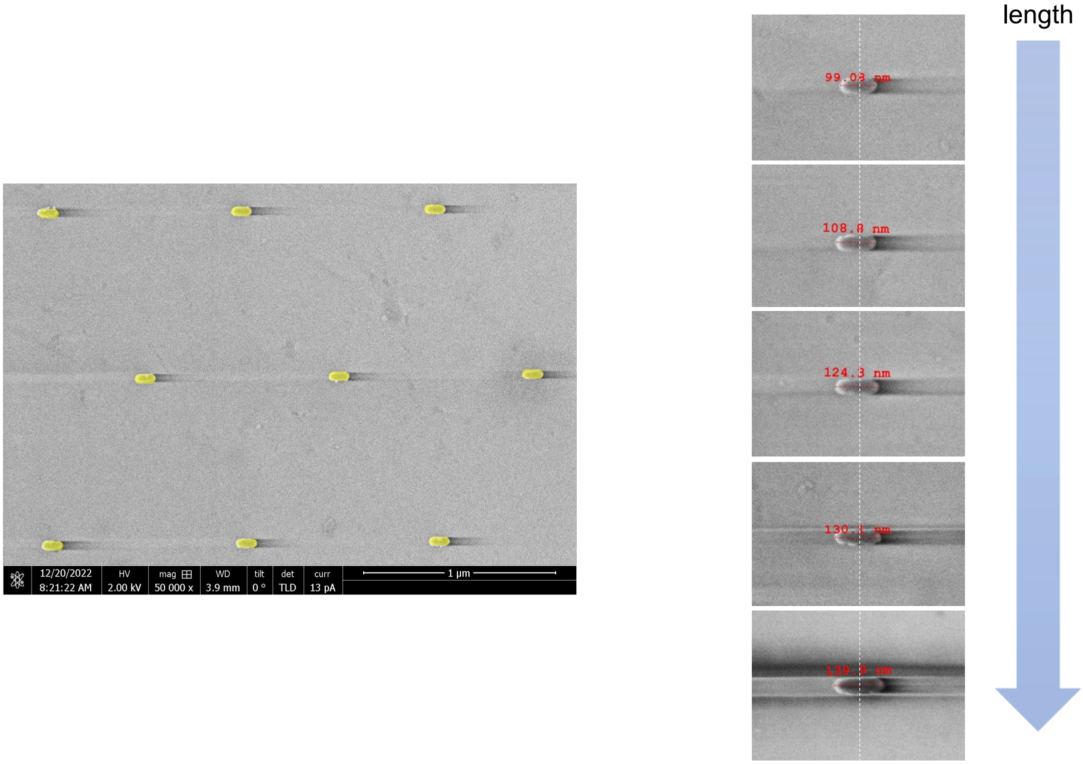

Figure S5 SEM image of an array of nanoantenna; in the left with a length of around120nm and periodicity of 1µm that we used for our study. In the right: SEM images of various nanoantenna fabricated with lengths ranging from 90nm to 130nm.

The nanoantennas were fabricated using electron beam lithography (EBL). Arrays of antennas with lengths ranging from 90nm to 130nm with a width of 30 nm were created to determine the optimal geometry for achieving a specific resonance. By placing the ZIF-8 film on top of these antenna, we aim to identify the array that resonate at a wavelength of 850nm when the film is in its fully adsorbed state.

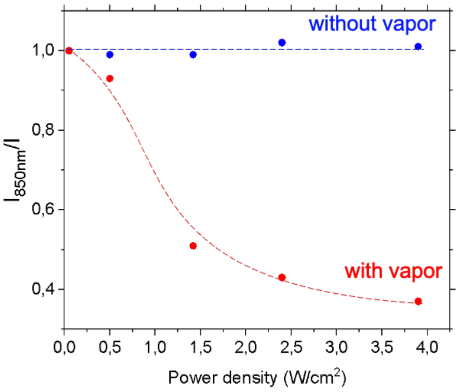

Figure S6 Evolution of the normalized scattering intensity at 850 nm as function of the incoming light intensity (the dotted lines are a guide for the eye).

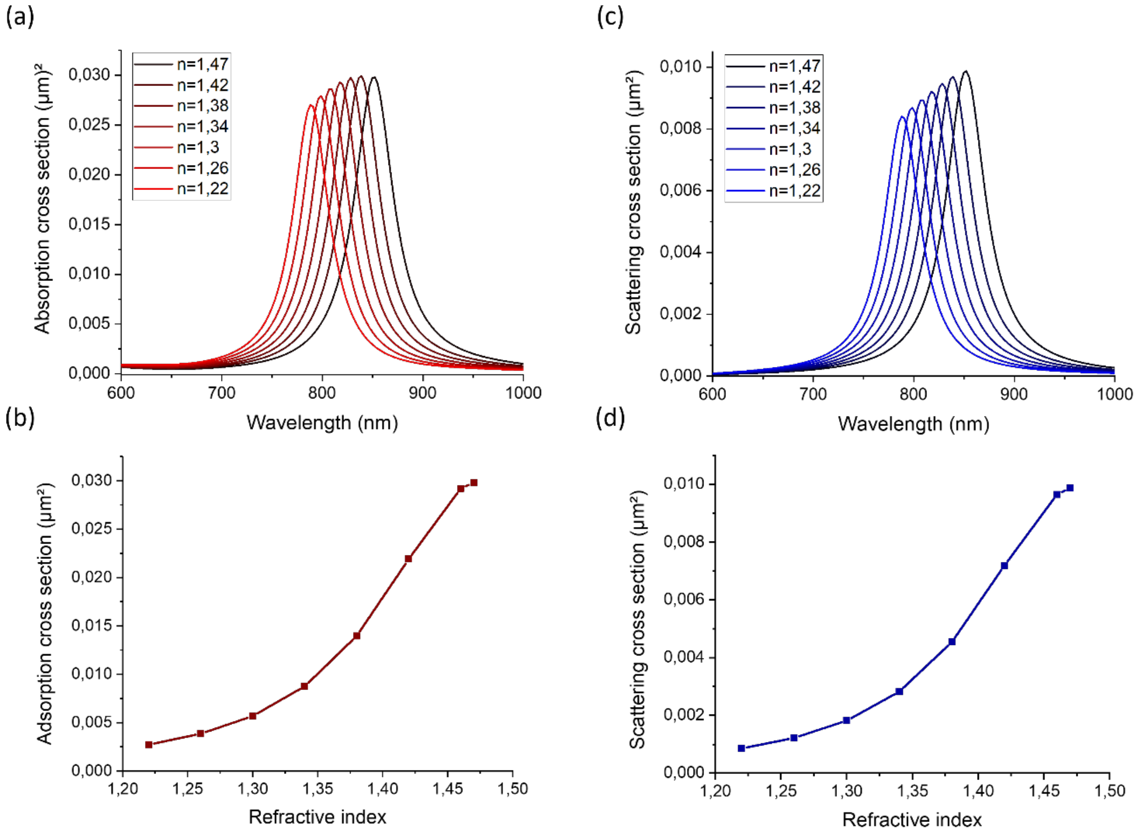

Figure S7 Simulation results indicating the correlation between the evolution of absorption and scattering at 850nm for varying refractive index of the MOF layer: (a) the adsorption cross-section of the gold nanoantenna as function of wavelength for different refractive index of the surrounding medium. (b) the adsorption cross-section as a function of the refractive index for a single wavelength of 850 nm. (c) the scattering cross-section of the gold nanoantenna as function of wavelength for different refractive index of the surrounding medium. (d) the scattering cross-section as a function of the refractive index for a single wavelength of 850 nm.

Estimation of the local temperature

To estimate the local temperature , we compared the reduction in normalized scattering at 850 nm, shown in Figure 4, with the normalized scattering at 850 nm as a function of temperature used as a calibration curve.[4] This curve was derived by simulation, starting from the experimental values of the refractive index of the ZIF-8 layer as a function of temperature shown in Figure 2(b).

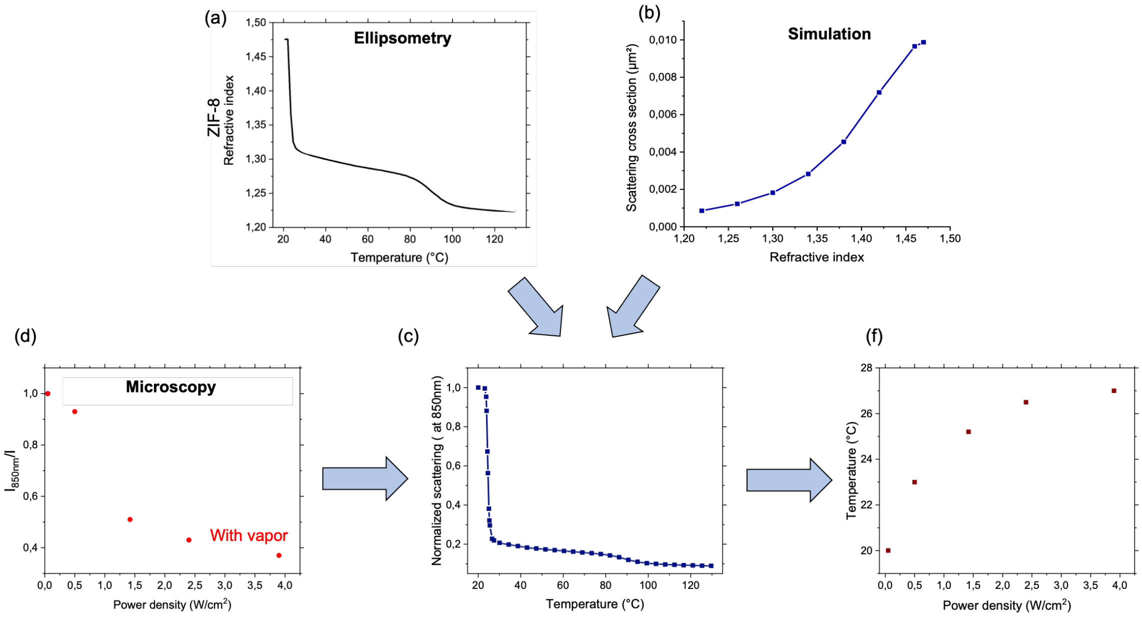

Figure S8 (a) Evolution of the refractive index as a function of the temperature in the presence of 0.7 P/P_0 of IPA for the desorption isobar measured by in situ ellipsometry. (b) scattering cross-section as a function of the refractive index for a single wavelength of 850 nm (c) Simulated evolution of the normalized scattering at 850nm as a function of the temperature (d) Evolution of the normalized scattering intensity at 850 nm as a function of the incoming laser power obtained experimentally ; (f) temperature evolution as a function of the laser power density

Dynamic analysis of independent antennas: experimental details and autonomous image analysis

The array of MOF-coated antennas was illuminated by an 850 nm laser through a condenser to ensure uniform illumination. The analysis was conducted with constant laser power over varying time intervals, progressively increasing and then decreasing the power. The results presented in the main text were obtained using a specific interval of laser powers (determined experimentally) at which the antennas exhibited the most prominent oscillations. Reducing the laser power too much prevented the evaporation of IPA around the antennas, inhibiting their oscillations. Conversely, using excessive laser power caused excessive heating of the antennas, which shifted the plasmonic peak and kept the area around them dry.

To analyze the oscillations, we captured images using the same microscope as for hyperspectral imaging, but with a standard camera instead of a hyperspectral detector. This camera is slightly sensitive to the 850 nm light scattered by each particle. Since the presence of MOF around the particles increases light scattering, as seen in Figure 5 of the main text, we were able to observe the filling and emptying cycles of each antenna, which were sufficiently spaced to be evaluated individually using optical methods. The array was not entirely uniform due to the detachment of some antennas during previous failed MOF deposition attempts. In fact, this irregularity was advantageous for image analysis, as it facilitated the identification of specific antennas during processing, minimizing errors.

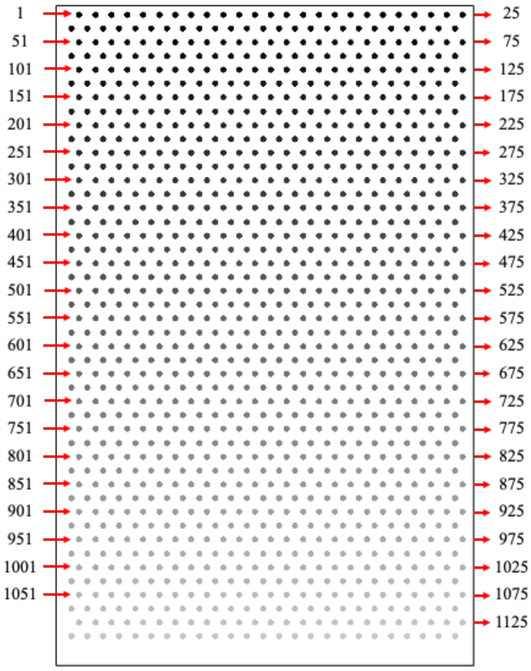

Figure S9 Supporting image to allow identify the antennas by gray-value. The numbering of each antenna is obtaining "reading" the set of antennas as normally done with a book. From left to right from top to bottom.

Figure S8 provides a "map" linking each particle to its corresponding number, enabling the reader to match the particle numbers shown in the titles of our plots with their actual locations in the image. The gray value of each circle corresponds to the antenna number. This facilitates a comparison of oscillations between nearby particles and may help identify patterns that might have been overlooked in our analysis. Based on our observations, the particles appear to behave as though they are isolated, not only optically but also "environmentally," meaning that the drying of MOF around one particle does not influence the behavior of neighboring particles. Consequently, we did not observe any collective behaviors or particles synchronizing in phase, as might be expected if the spacing between them were reduced. This will be explored in future publications.

To begin our analysis, we stabilized the "raw" data, which consisted of a series of images captured every second, using a custom tracking and stabilization algorithm. The algorithm translates the images by the number of pixels determined by the tracking process. In cases where the antennas' XY movement was too rapid, manual tracking was applied for a few frames.

No Z-axis stabilization was performed since we only had images of a single focal plane. However, the focus was maintained consistently throughout the experiment.

The luminosity of the scattered light from each particle was analyzed using a straightforward algorithm based on the "Plot Z-Axis Profile" function in ImageJ, as described below. This macro allowed us to capture both the average luminosity and the individual luminosity of each antenna in a single plot. By using the "data" button and selecting the "copy all data" option, we exported the scattering values as numerical data into an Excel file.

The data, in .xlsx format, were then imported into a Mathematica notebook, where they were organized based on laser power, creating lists of lists, each containing data for all particles with *n* time points (depending on the duration the specific laser powers were applied, with transient periods between power changes excluded). The scattered intensity values were then fitted to a straight line, and the value of this linear fit was subtracted from the intensity data. This step was performed to eliminate any drift in intensity, thereby making the values comparable by centering them around zero.

To continue the analysis, the oscillation data were normalized by dividing by the absolute value of the maximum or minimum of the linearly adjusted intensities, restricting the oscillations between 1 and -1. The normalized time series were then processed using a numerical function in Mathematica to compute their Fourier transform. Since the time series had different numbers of data points, the frequency plots had varying spacings between frequency values, calculated as ½*2π/T, where *T* is the total time of data acquisition at a given laser power.

XY Tracking Algorithm explanation: The tracking algorithm processes a video that includes a few antennas to minimize computation time. The user selects one antenna, which is assumed to remain within the video frame throughout. Since the antenna's position corresponds to the maximum brightness of its scattered light, the algorithm isolates it by multiplying each frame with a secondary frame that has a Gaussian distribution of brightness centered on the maximum from the previous frame. This helps to avoid tracking errors caused by unwanted optical effects or sudden movements. For each multiplied frame, the position of the maximum is identified and recorded. The displacement of the maximum in each frame, relative to the first frame, is then calculated and used in the subsequent stabilization process. The algorithm was implemented as an ImageJ macro, and its code is provided here. The program generates the list of the displacements in the "Log" of imageJ. Expected values of "displacements" are between -2 and +2. This list has to be saved in a text file (.txt) that has to be given as input to the "moving" algorithm.

XY Tracking Algorithm ImageJ macro Code:

```
ImgID=getImageID();
rename("Source");
Name=getTitle();
getDimensions(width, height, channels, slices, frames)

MaxIntX=101; // X-position antenna we want to track
MaxIntY=127; // Y-position antenna we want to track
Sigma=6;     // "radius" of the gaussian
FindSpot(MaxIntX,MaxIntY,slices,Sigma,Name);

function FindSpot(MaxIntX,MaxIntY,slices,Sigma,Name) {
        MaxX=MaxIntX;
        MaxY=MaxIntY;
        for (n = 1; n < slices+1; n++) {
                //print("ok", n);
                selectWindow(Name);
                setSlice(n);
                run("Make Substack...", "  slices="+n);
                DrawGaussian(MaxX,MaxY,Sigma,width,height);
                imageCalculator("Multiply", "Substack ("+n+")","Gauss");
                selectWindow("Substack ("+n+")");
                run("Find Maxima...", "noise="+(width+height)+" output=List");
                MaxX = getResult("X",0 );
                MaxY = getResult("Y",0 );
                if (n==1) {
                        MaxXstart=MaxX;
                        MaxYstart=MaxY;
                }
```

```
			print(MaxX-MaxXstart,",", MaxY-MaxYstart);
			selectWindow("Substack ("+n+")");
			close();
			selectWindow("Gauss");
			close();
		}
}

function DrawGaussian(CenterX,CenterY,Sigma,width,height) {
		newImage("Gauss", "32-bit black", width, height, 1);
		for (i = 0; i < width; i++) {
			for (j = 0; j < height; j++) {
				//ValInput= getPixel(i, j);
				ValInput=1;
				PosX= i-CenterX;
				PosY= j-CenterY;
				Value =ValInput*exp(-1*(PosX*PosX+PosY*PosY)/2/Sigma/Sigma);
				setPixel(i		, j, Value);
			}
		}
}
```

XY Translate Algorithm explanation: The algorithm simply takes the values obtained by the previous algoritm saved in the .txt file (please put the right folder in the macro) and uses them in the “Translate” function of ImageJ. It proceeds in doing so slice by slice (timepoint by timepoint).

XY Translate Algorithm ImageJ macro Code:

```
ImgID=getImageID();
rename("Source");
Name=getTitle();
FilePathName="	"; //put filepath of your .txt file with obtained by the previous algoritm

print("\\Clear");
TranslateXY(FilePathName);

function TranslateXY(FilePathName) {
		getDimensions(width, height, channels, slices, frames);
		RetArrayXY = parse_moduli(FilePathName);
		deltind=RetArrayXY.length/2;
		if (frames==1) {
			frames=slices;
			slices=1;
		}
		for (i = 1; i < slices*frames+1; i++) {
			setSlice(i);
			temp=floor((i-1)/slices);
			run("Translate...", "x=" + (-RetArrayXY[temp]) +" y="+ (-RetArrayXY[temp+deltind])
+ " interpolation=None slice");
			print(i,temp,-RetArrayXY[temp],-RetArrayXY[temp+deltind]);
		}
}

function parse_moduli(FilePathName) {
//parse the moduli file
```

```
	moduli_file = File.openAsString(FilePathName);
	moduli_lines = split(moduli_file, "\n");
	moduliX = newArray(moduli_lines.length);
	moduliY = newArray(moduli_lines.length);
	for (m=0; m<lengthOf(moduli_lines); m++){
		moduli_n= split(moduli_lines[m], ";,");
		moduliX[m] =   parseFloat(moduli_n[0]);
		moduliY[m] =   parseFloat(moduli_n[1]);
	}

	RetArray=Array.concat(moduliX,moduliY);
	return RetArray;
}
```

Scattering intensity extraction algorithm: The intensity extraction algorithm processes the video, which contains hundreds of antennas arranged in hexagonal symmetry. The "Plot Z-Axis Profile" function is applied to each antenna, following the numbering provided in Figure SX. It is crucial to accurately define the distance between the antennas to ensure that the analysis is correctly centered on each one. Additionally, the reader has to specify the radius when drawing the circle for the "Plot Z-Axis Profile" that should be approximately half the radius of the antenna's scattering zone to minimize the impact of XY vibrations, which can still affect the video at the pixel level.

After performing the "Plot Z-Axis Profile," the program draws a circle over the analyzed area, allowing the user to verify that the particle has been properly centered. All intensity profiles are compiled into a single plot, and the data can be exported into Excel for further analysis. By setting to 1 the value "drawing" it is possible to ask the program to fill the circles with a gray value corresponding to the numbering of the antenna (which will be useful for the following analysis with Mathematica). This algorithm was implemented as an ImageJ macro, and the code is provided here.

Scattering intensity extraction ImageJ macro code:

```
ImgID=getImageID();
rename("Source");
Name=getTitle();

FirstX= -1; // X position of the upper left corner of the oval corresponding to the first particle
FirstY= -10; // Y position of the upper left corner of the oval corresponding to the first particle
OvalRad=10; // radius of the oval used to analyze the particles.
distance_particles=23.24;

zprofiling=1; // deactivate (=0) this to oly perform the numbering of the antennas
drawing=1; // activate (=1) this if you want to draw a circle with color values equal to the antenna
number after having done the Z profile. Useful for double checking the antenna number

RefX=FirstX;
RefY=FirstY;
if (zprofiling == 1) {
	run("Plot Z-axis Profile");			// the first profile takes the average of the full image and
produces a plot with the name Source-0-0 that is useful to be selected
}
setBatchMode(true);

N_particles_X=25;  // N of particles in the X direction
N_particles_Y=30;  // N of particles in the Y direction
```

```
counter=0;

for (j = 1; j < N_particles_Y+1; j++) {
    CorrX=0.5*distance_particles;
    if ((j/2) == floor (j/2)) {
        CorrX=0;
    }
    for (i = 1; i < N_particles_X+1; i++) {
        CenterOvalX=FirstX+i*distance_particles;
        CenterOvalY=FirstY+j*distance_particles*sqrt(3/4);
        if (zprofiling == 1) {
            print(CenterOvalX+CorrX,CenterOvalY);
            MeasZprof (CenterOvalX+CorrX,CenterOvalY,OvalRad,Name,ImgID);
        } else {
            makeOval(CenterOvalX-OvalRad/2+CorrX,           CenterOvalY-OvalRad/2,
OvalRad, OvalRad);
            if (drawing == 1) {
                Color.set(j*N_particles_X+i)
                Color.set(counter)
                run("Fill", "slice");;
            }
        }
        run("Draw", "slice");
        counter=counter+1;
    }
}
setBatchMode(false);

function MeasZprof (PosXset,PosYset,Diam,Name,ImgID) {
    PosX=PosXset-Diam/2;
    PosY=PosYset-Diam/2;
    selectImage(ImgID);
    makeOval(PosX, PosY, Diam, Diam);
    run("Plot Z-axis Profile");
    selectWindow(Name+"-"+0+"-"+0);
    //print(PosX,PosY);
    Plot.addFromPlot(Name+"-"+floor(PosX)+"-"+floor(PosY), 0);
    selectWindow(Name+"-"+floor(PosX)+"-"+floor(PosY));
    close();
    selectImage(ImgID);
    makeOval(PosX, PosY, Diam, Diam);
    run("Draw", "slice");
}
```

Fourier transform and data selection: The scattered intensity values from each antenna are stored in an Excel file, which is opened in Mathematica for analysis. The first row and the first two columns of the file are discarded, as they contain the antenna names, slice numbers, and the average intensity of the full image, respectively. The remaining data are split into different series, each corresponding to a different laser power, allowing for easier data manipulation. Data collected while the laser power was being changed are excluded from the analysis.

Each data series, which contains a different number of data points (collected at one-second intervals), is normalized to the corresponding laser power. Then, any linear trend or average value in the data—due to small variations in focusing or camera efficiency over time—is removed. This step does not affect the Fourier transform of the antenna oscillations. After removing the linear trend, the data are normalized to the maximum or minimum value within each series, resulting in data values that range between -1 and 1.

The Fourier transform algorithm is then applied to the normalized data. The output provides the intensity of the oscillations at different frequency intervals for each series. It is important to note that the resolution of the X-axis (frequency) in the Fourier transform is determined by the inverse of the sampling rate (in this case, 1 image per second) multiplied by the inverse of the total number of data points collected. This means that series with longer observation periods offer better frequency resolution. For more detailed understanding of the discrete Fourier transform algorithm, readers should refer to Mathematica and other related documentation.

In “ALL-Antennas.doc”, we present the time-dependent normalized scattered intensity (black) alongside its Fourier transform (red). The axes are color-coded to correspond with each value. For some particles, distinct peaks are visible in the Fourier transform, while for others, this is not the case. This analysis enables straightforward identification of the oscillating antennas sorting them as respect to the frequency, light power...As an example, in Figure S10, 11 and 12 we show the top 9 particles oscillating at different frequency under laser power 2.4W cm^{-2}.

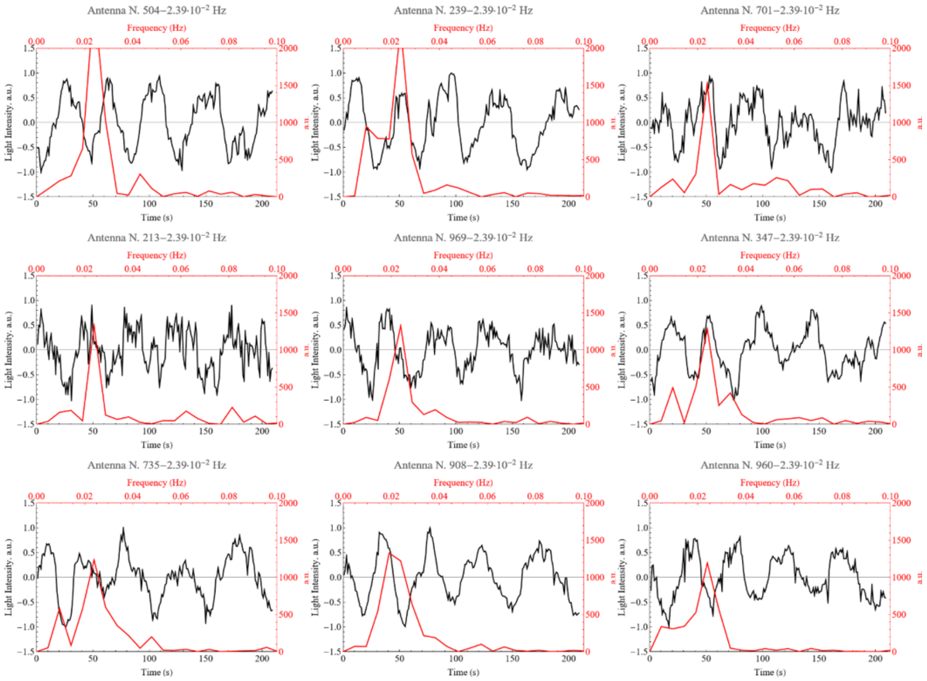

Figure S10 Best particles oscillating at 2.39 10^{-2} Hz under laser power 2.4W cm^{-2}

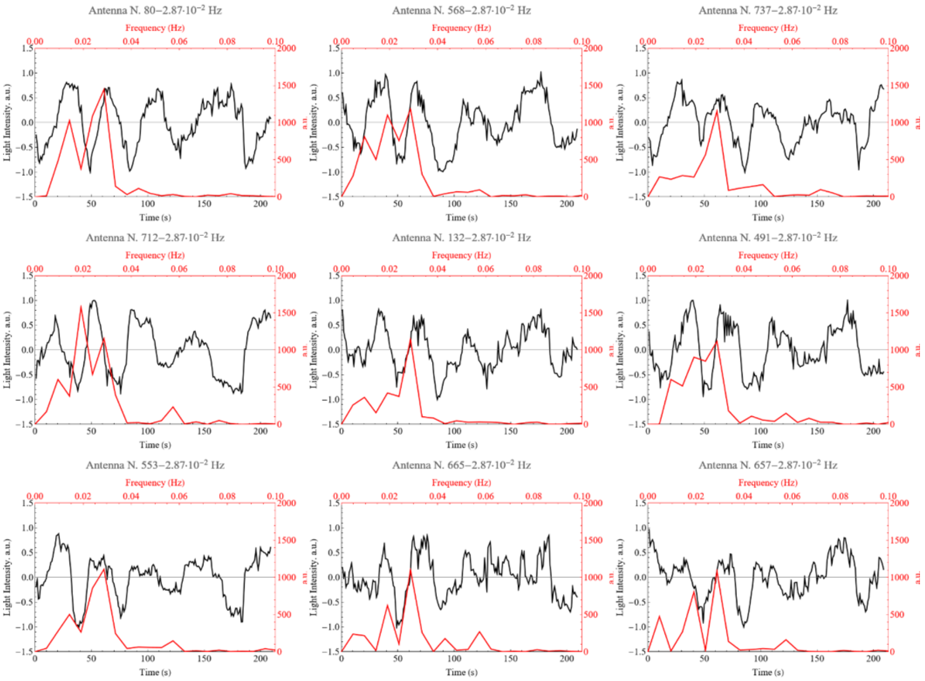

Figure S11 Best particles oscillating at 2.87 10^{-2} Hz under laser power 2.4W cm^{-2}

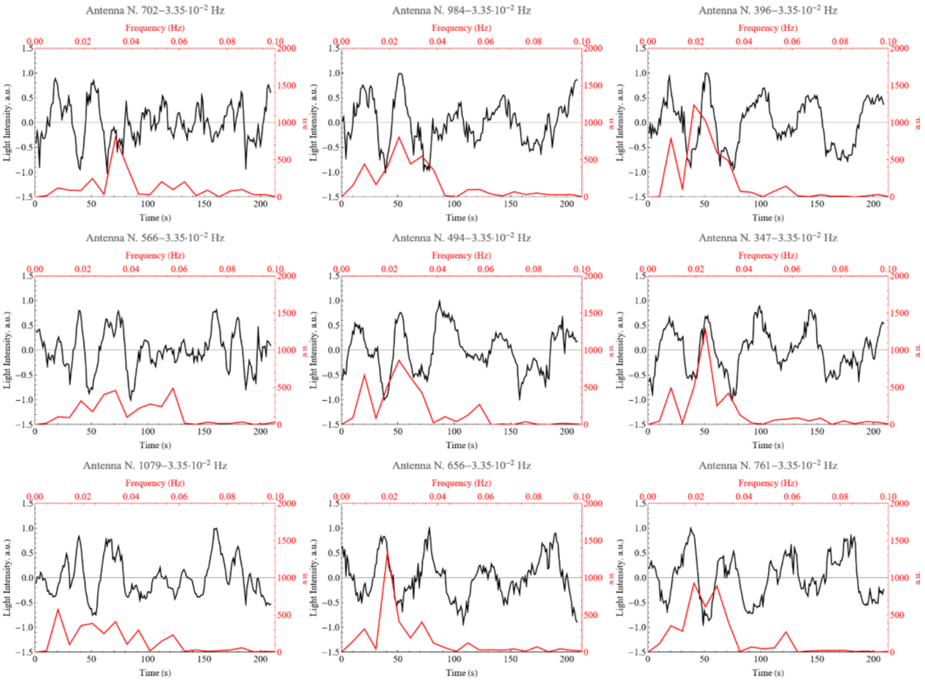

Figure S12 Best particles oscillating at 3.35 10^{-2} Hz under laser power 2.4W cm^{-2}

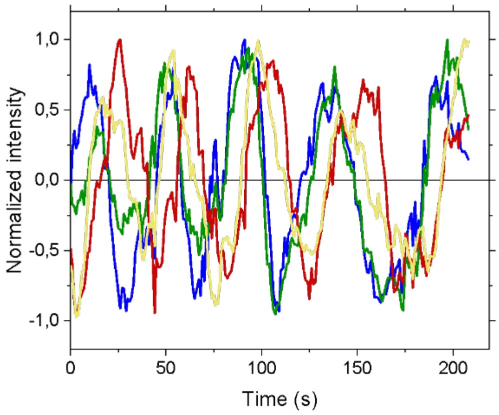

Figure S13 Scattering evolution of four antennas not located close to each other exhibiting a similar frequency but not in phase

Thermal simulation

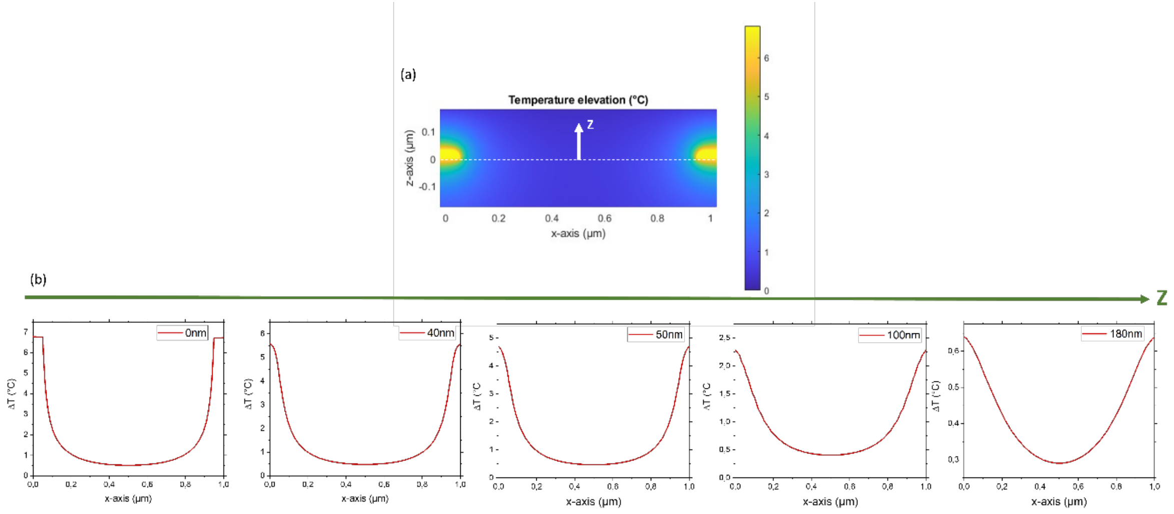

Figure S14 (a) Heat map of temperature elevation for two nanoantenna with a periodicity of 1µm, illustrating the temperature distribution along the x-axis (0 to 1µm) and the z-axis (0 to 0.2 µm). The temperature distribution in a 2D plot, where the color intensity indicates the level of temperature elevation. (b) Series of line plots indicating the temperature distribution along the x-axis for different z-values (z= 0, z=40nm, z= 50nm, z=100nm and z=180nm respectively). Each plot shows the temperature profile through the two nanoantennas

For the thermal simulation, we accounted for the power of heat generation Q_v (power density) inside the nanostructure. This power is directly proportional to the Absorptivity A calculated theoretically , P_L laser power or the incident light and the nanoantenna volume V, with:

$$Q_v = \frac{A * P_L}{V}$$

We specified the thermal conductivity values for the materials in each domain. For the gold antenna, the thermal conductivity was set to 318 W/m·K. The simulation was carried out in a 3D domain with the following properties:

Material	Thermal Conductivity (W/m·K)	Specific Heat Capacity (J/m³·K)	Mass Density (kg/m³)
Gold (Au)	318	2.472×10^{6}	19.32×10^{3}
Air	0.0259	1.22×10^{3}	2.2×10^{3}
Substrat	1.6478	1.6478×10^{6}	2.2×10^{3}
ZIF-8 Film	0.3	8.84×10^{5}	0.94×10^{-3}

The thermal conductivity values for ZIF-8 were examined in the study of Wei et al,[5] it was revealed that thermal conductivity of ZIF-8 increases linearly from 0.165 W/m.K to 0.319 W/m.K during ethanol adsorption, where ethanol has a thermal conductivity of 0.17 W/m.K. Given that the thermal conductivity of isopropanol is 0.15 W/m.K, we can estimate the increase in thermal conductivity of ZIF-8 when isopropanol is adsorbed by applying the linear relationship, we estimate that the thermal conductivity could increases to approximately 0.3 W/m.K. We only considered a convective heat transfer at the top and the bottom of studied domain where the convective coefficient is equal to 20 ($W/m^2 \cdot K$). The volumetric heat sources of gold nanoantenna were applied using a Gaussian distribution to represent the heat generation more accurately and equal to 1.5×10^{-2}. The initial temperature was set to 293.15 K. Boundary conditions were established using Dirichlet conditions to simulate realistic constraints with (Dirichlet ($T = T_0$) = 293.15)

We then computed the global matrices for the thermal system and incorporated the heat sources. The stationary thermal system evolution was solved to obtain the temperature distribution within the nanoantenna array.

1. Amyar, H. *et al.* In situ infrared spectroscopic ellipsometry as a tool to probe the formation of sol–gel based mesoporous films. *Journal of Sol-Gel Science and Technology* 1–12 (2023).
2. Dalstein, O. *et al.* Nanoimprinted, Submicrometric, MOF-Based 2D Photonic Structures: Toward Easy Selective Vapors Sensing by a Smartphone Camera. *Advanced Functional Materials* **26**, 81–90 (2016).
3. Johnson, P. B. & Christy, R.-Wjp. Optical constants of the noble metals. *Physical review B* **6**, 4370 (1972).
4. Chehadi, Z., Boissière, C., Chanéac, C. & Faustini, M. Nanoconfined water vapour as a probe to evaluate plasmonic heating. *Nanoscale* **12**, 13368–13376 (2020).
5. Wei, W., Huang, J., Li, W., Peng, H. & Li, S. Impacts of ethanol and water adsorptions on thermal conductivity of ZIF-8. *The Journal of Physical Chemistry C* **123**, 27369–27374 (2019).